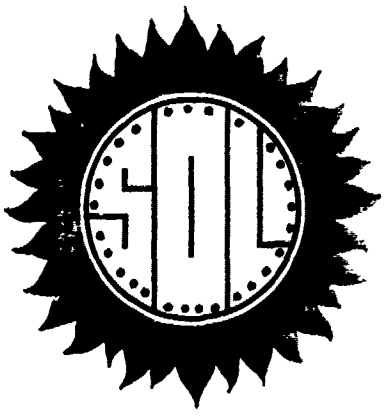

Systems
Optimization
Laboratory

Barrier Methods for Large-Scale Quadratic Programming

by
Dulce B. Ponceleón

TECHNICAL REPORT SOL 91-2

June 1991

Department of Operations Research
Stanford University
Stanford, CA 94305

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

Barrier Methods for Large-Scale Quadratic Programming

by
Dulce B. Ponceleón

TECHNICAL REPORT SOL 91-2

June 1991

Research and reproduction of this report were partially supported by the National Science Foundation Grant DDM-8715153 and the Office of Naval Research Grant N00014-90-J-1242.

A mi mami:
Por su eterno apoyo incondicional

BARRIER METHODS FOR LARGE-SCALE QUADRATIC PROGRAMMING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Dulce B. Ponceleón
June 1991

Abstract

Barrier Methods for Large-Scale Quadratic Programming

by

Dulce B. Ponceleón
October 1990

We present several new algorithms for solving the general large-scale quadratic programming (QP) problem.

A feature of QP problems is the presence of linear inequality constraints, which introduce a combinatorial aspect to the problem. Currently the most common approach to solving QP problems is to apply active-set methods, in which only some of the inequalities are used to produce a search direction at each stage. The combinatorial element is therefore inherent. As problems become larger, there is a potential for an excessive number of iterations and consequent inefficiency.

In contrast, we use the now familiar barrier-function approach, which circumvents the combinatorial aspect by introducing a barrier transformation involving all of the inequalities. The barrier term enforces satisfaction of the inequalities by modifying the objective function. The transformed problem is solved by a modified Newton method applied to the nonlinear equations defining feasibility and optimality.

The main computation at each iteration of the new algorithms is the solution of an indefinite system of linear equations. Barrier methods are known to lead to ill-conditioned systems. However, we show by a special sensitivity analysis that the particular manner in which we have formulated the barrier transformation leads to ill-conditioning that is benign.

We address the many details that need to be resolved in order to define an efficient algorithm for solving large-scale QP problems. A specific barrier algorithm has been implemented, with linear programming (LP) included as a special case. Numerical results are presented for a set of sparse QP test problems. A feature of the implementation is that its efficiency does not depend on whether the solution is near or far from a vertex of the feasible region.

Acknowledgements

I am indebted to *many* people who have made this dissertation possible. I have been lucky to be closely associated with two departments at Stanford, which has meant I have met and been influenced by more people than usual. I have had the benefit of being part of two student circles.

In the Computer Science Department I am very grateful to all those who made the *NA hall* such a cheerful and stimulating place to work. In addition to the common technical interests, the nights behind the terminal, the gripes about the system, the jokes and the popcorn dinners, their friendship greatly contributed to my Stanford experience.

In the Department of Operations Research I was fortunate to share in the camaraderie and many technical discussions that were relevant to my thesis.

I would like to thank Professor Richard Cottle for his bibliographical assistance and Dr. Margaret Wright for reading the final manuscript and suggesting a number of constructive changes.

I am grateful to Professor Philip Gill for his help in the early stages of my thesis. Regretfully his departure to San Diego prevented me benefiting more from his expert knowledge of nonlinear programming.

I would like to thank the members of my committee. I am very grateful to Professor Joe Oliger for his financial support over several summers, for serving on my oral committee and for promptly reading my thesis. His diligence has made me feel that if I ever needed it, he would be there to help.

I would like to thank Professor Gene Golub for his generosity when I arrived at Stanford and for teaching me the vital elements of numerical linear algebra. Gene is largely responsible for the good spirit of the NA group in the CS department. His influence in bringing many distinguished visitors to Stanford contributes greatly to the education of the student body in general and to my own education in particular.

Special thanks are due to Professor Michael Saunders whose contribution to this thesis goes way beyond that normally associated with a member of the oral and reading committees. Michael's interest in my work and his patience when giving me help are much appreciated. His involvement has contributed greatly to the finished thesis. I am particularly grateful for his tolerance and conscientiousness in times of stress. Despite my being difficult to help at times, Michael persevered.

Finally, I am deeply indebted to my advisor Professor Walter Murray. His generosity with both his time and ideas knows no bounds. He even has the good manners

to forget what he has done. Walter treats students as equals and often does the kind of chores for a student that students normally do for their professors. He also does everything in a cheerful spirit. It is not uncommon for Ph.D. candidates to become despondent from time to time and I was no exception. However, on such occasions I always knew that the unfailing enthusiasm and optimism of my advisor would help me get through. I can do no better than to end by quoting two of his former students: "...it has been a rare privilege to be your student".

This work was partially supported by the Venezuelan Government Institution CONICT (Concejo Nacional de Investigación Científica y Tecnológica), the Office of Naval Research Grant N00014-90-J-1242, and the National Science Foundation Grant ECS-8715153.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Statement of the Problem

In this thesis we present several new algorithms to solve the *general* quadratic program (QP) of minimizing a quadratic objective function subject to linear constraints on the variables. Quadratic programming problems are among the best understood areas of optimization, with algorithms for them dating back to the 1950's. Many mathematically equivalent formulations of the problem are possible, and the choice depends mainly on the context. It will be seen later that a very specific form of the problem is crucial to the barrier algorithms presented in this dissertation, but initially we shall consider quadratic programs in the following form:

$$\boxed{\begin{array}{ll} \underset{x\in\Re^n}{\text{minimize}} & Q(x) = c^Tx + \frac{1}{2}x^THx \\ \text{subject to} & Ax \geq \beta, \end{array}} \qquad (1.1.1)$$

where the *Hessian matrix* H of the quadratic function is an $n \times n$ symmetric matrix, c is an n-vector, A is an $m \times n$ matrix, and β is an m-vector. We are particularly interested in the case where the matrices A and H are *large* and *sparse*, and we include the general linear programming (LP) problem arising when $H = 0$.

We shall assume at least one bounded solution to (1.1.1) exists, say x^*. When H is positive semidefinite, the problem is termed a *convex* QP. A useful property of a convex QP is that a *local* minimizer x^* of a convex QP is also a *global* minimizer. When H is positive definite, x^* is unique. When H is indefinite, more than one local solution may exist. Moreover, some of the local solutions may not be global. Naturally these characteristics of the solution have an important impact on the design of an algorithm.

The problem of finding a global minimizer of a general optimization problem is very difficult and the indefinite quadratic case is no exception. It can be shown that finding a global minimizer of an indefinite QP is equivalent to finding an integer solution to a linear program, a known difficult problem. Hence, most algorithms do not attempt to find more than a local minimizer. Indeed, even the verification that some point is a *local* minimizer of a general quadratic program is under some circumstances

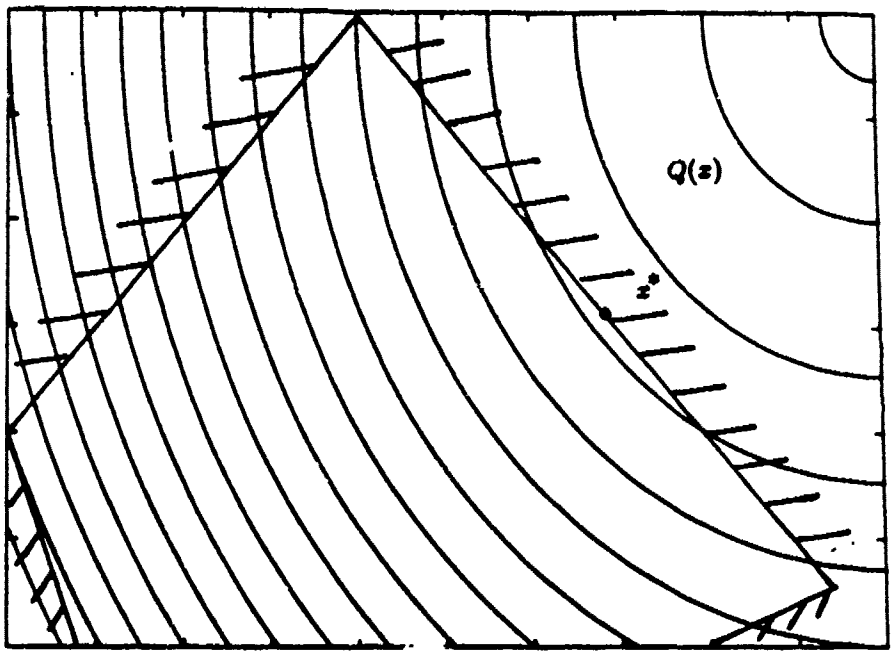

Figure 1.1: Con QP Problem

an NP-hard problem—see Contesse [Con80], Murty and Kabadi [MK87] and Pardalos and Schnitger [PS88].

Figure 1.1 shows the contours of a strictly convex quadratic function and the feasible region defined by the linear constraints. The shaded area corresponds to the infeasible region. In this example the solution x^* is unique, i.e. it is a strong global minimizer. Figure 1.2 shows the contours of a nonconvex quadratic function. This function has two isolated local minimizers, only one of which is a global minimizer.

In contrast with a typical linear programming problem, the solution to either problem does not lie at a vertex of the feasible region. Notice that if the minimizer x^* was in the interior, it would imply that x^* was also a minimizer of the unconstrained problem, or equivalently, the linear constraints would not play any role. In general, the minimizer of a constrained QP is expected to lie on a boundary of the feasible region, although not necessarily at a vertex.

For large-scale problems, very few practical, general-purpose algorithms are currently available. Almost all current methods for QP whether for large or small problems are of a type known as active-set methods. In this thesis we describe algorithms for QP that are not active-set methods. In developing a large-scale QP algorithm of a radically different nature, we avoid some of the fundamental drawbacks present in active-set methods. Certain *different* difficulties exist, but we anticipate that at least for some problems the balance will favor the new approach.

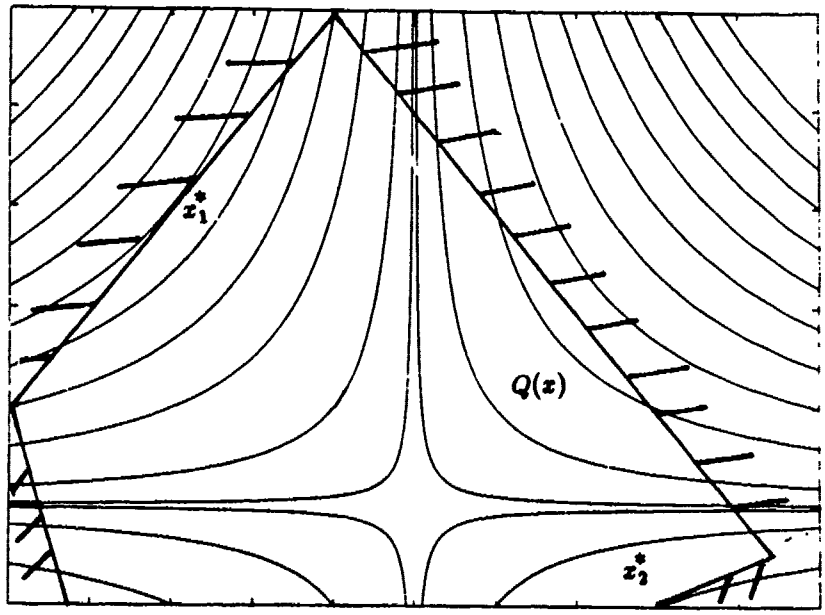

Figure 1.2: Non-convex QP Problem

1.2 Applications

Quadratic programs arise from many diverse applications. For instance, the well known PILOT problem [SMD86] is a large-scale dynamic model of the U.S. economy that synthesizes representations of all sectors of the economy in a general equilibrium context. PILOT simulates economic interactions among sectors by determining market-clearing prices and quantities for all commodities over time. The Hessian matrix is diagonal and semidefinite. Approximately one percent of the variables contribute to the quadratic term.

Portfolio Optimization in the business world constitutes another source of quadratic programs. Specific applications include asset allocation (such as stocks and bonds), and optimizing risk-return tradeoffs assuming superior investment judgement. Many investment advisory firms and pension-plan sponsors today routinely compute mean-variance efficient portfolios as part of the portfolio allocation process. Given that investment companies use large-scale portfolio models to trade *billions of dollars*, the development of a robust large-scale QP algorithm is of paramount importance.

As well as being important in their own right, quadratic programs often appear as subproblems within algorithms for nonlinear programming. Our interest in sparse quadratic programs arises in part from the desire to apply sequential quadratic programming (SQP) methods to large nonlinearly constrained problems. As the name suggests, each iteration of an SQP method involves solving a quadratic programming

subproblem. These methods are regarded as state-of-the-art methods for the solution of nonlinear problems. The overall performance depends critically on the efficient solution of the QP subproblems.

Naively, one might think that to develop an SQP method for solving nonlinear programs, it would suffice to have a *black box* for solving the quadratic subproblems. Unfortunately, this is not true even in the dense case; see Gill *et al.* [GMSW85] for a detailed discussion of the difficulties involved in adapting active-set QP algorithms to the SQP environment.

1.3 Historical Background

Given the importance of quadratic programming, it is not surprising that many algorithms have been proposed. Some of these were derived as extensions of the simplex method for linear programming, in the sense that they involved pivot operations on a simplex-type tableau. These include Wolfe's simplex method for QP [Wol59] and some of the algorithms for linear complementarity problems—notably the principal pivoting method of Cottle and Dantzig [CD68,Cot68,Cot89] and Lemke's method [Lem62].

Beale's method [Bea55,Bea59] was one of the more successful early approaches. It has the feature of tending to be more efficient if $Q(x)$ is linear in most of the variables (i.e. if H has low rank). In many early QP algorithms it was necessary for H to be positive definite and such algorithms were therefore restricted in their application. In particular, they were unsuited to problems derived as extensions to LP models—a common source of QP's.

The equality-constrained quadratic program (EQP) constitutes the simplest QP. In this case, the constraints are of the form $Ax = b$. A distinctive feature of an EQP is that its solution, if it exists, can be obtained by solving a *single* system of linear equations (known as the *Karush-Kuhn-Tucker* or KKT system) involving the matrix

$$K = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}. \tag{1.3.1}$$

No iterative procedure is needed. A variety of numerical methods have been developed to *compute* the solution of EQP. The major differences among them arise from the numerical techniques used to solve the system of linear equations. The method of choice depends on the size and representation of the problem.

One approach is to use a factorization of K itself—sometimes referred to in the literature as *Lagrangian* or *Karush-Kuhn-Tucker* methods. Other approaches, the so-called *projection* methods, involve breaking down the system into smaller and simpler systems. The factorizations needed are then smaller than in Lagrangian-type methods, but less likely to be sparse.

The constraint matrix A can be viewed as defining two complementary subspaces: the null space of vectors orthogonal to the rows of A, and the range space of vectors spanned by the rows of A. From this observation, two categories of projection methods have been developed, namely *null-space* methods and *range-space* methods. In many cases, the work required to solve EQP is directly proportional to the dimension of the corresponding subspace.

EQP methods are fundamental for solving subproblems that arise in *active-set* methods for the inequality-constrained quadratic program (IQP). Unlike an EQP, an IQP can be solved only by iteration. In the IQP case, a key piece of information is unknown—namely, the set of constraints that hold with equality at the solution. Such constraints are termed the *active set* of constraints at the solution or simply the active set. If the active set were known *a priori*, the solution of an IQP could be computed directly by solving a *single* EQP problem (ignoring the inactive constraints).

In the 1970's and 80's a number of sophisticated active-set methods were developed. Such methods are so-called because they maintain a *prediction* of the active set, called the *working set*. This is a linearly independent set of constraints that are satisfied exactly at the beginning of each iteration. Information is gathered at the current iterate to allow changes in the working set to improve the prediction. Iterations proceed until the active set is identified. It is common for any given constraint to enter and leave the working set many times. Although the working sets may not repeat, the number of possibilities is a combinatorial function of the number of constraints and variables. Early active-set methods are those due to Dantzig and Wolfe [Dan61], Fletcher [Fle71], and Murray [Mur71a]. More recent methods are due to Gill and Murray [GM78], Powell [Pow81], Goldfarb and Idnani [GI83], and Gill *et al.* [GMSW84c].

Naturally, some of the IQP methods are classified according to the type of method they use to solve the EQP subproblems. For example, the methods of Murray [Mur71a], Gill and Murray [GM78], and Bunch and Kaufman [BK80] are null-space methods, and are more efficient when the number of constraints in the working set is close to n, since the dimension of the null space is then relatively small. Range-space methods increase in efficiency as the number of constraints in the working set decreases. For an example of a range-space method, see Gill *et al.* [GGM*84].

We emphasize that under certain conditions (for example, when H is positive definite), many active-set methods for solving IQP are *mathematically equivalent*, in the sense that they calculate identical sequences of iterates when applied to the same problem with the same initial working set. (See Djang [Dja79], Cottle and Djang [CD79] and Best [Bes84].)

Approaches that treat the indefinite system (1.3.1) directly, may be especially appropriate for sparse problems; see Duff and Reid [DR82], Forsgren and Murray [FM90] and Gill *et al.* [GMSW84b,GMSW87].

An important class of active-set methods are the so-called inertia-controlling algorithms for QP (ICQP methods). Such methods restrict the change that can take place between successive working sets, placing a strict control on the degree of indefiniteness of the reduced Hessian matrix (a measure of nonconvexity). Lagrangian, null-space and range-space methods may all be ICQP methods. A complete description of ICQP methods is given by Gill *et al.* [GMSW88].

Let A_k denote the matrix of constraints defining the working set at iteration k. All active-set methods can be viewed as solving a sequence of linear systems involving matrices of the form

$$K_k = \begin{pmatrix} H & A_k^T \\ A_k & 0 \end{pmatrix},$$

i.e., a sequence of EQP problems. A feature of most active-set methods is that A_k differs little from A_{k+1}. Advantage may be taken of this feature by utilizing matrix factorization updating techniques [GGMS74].

While active-set approaches are well suited to many problems, the potential exists for the number of iterations to be extremely large even for problems of moderate size. In fact, in the simplex method for linear programming [Dan63]—an active-set method—the number of iterations in the worst case can rise exponentially with the number of constraints [KM72]. While it is well known for the simplex method that this worst case is rarely attained, the combinatorial aspect of active-set methods implies that the number of iterations often does grow significantly with the number of constraints. The wish to circumvent this feature of active-set methods is what motivates our interest in barrier methods, just as it motivated Karmarkar's approach to large-scale linear programming [Kar84]. Another reason is the difficulty in extending matrix updating procedures to the large-scale case.

1.4 Barrier-function Methods

The use of barrier methods for solving nonlinear optimization problems dates back to the late 1950's, and was subsequently well established in the 1960's by Fiacco and McCormick [FM68]. It can be appreciated from the preceding discussion that the combinatorial element in active-set methods arises because of the presence of *inequality* constraints. A barrier method eliminates inequality constraints by transforming the objective function. For example, the problem

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad F(x) \\ &\text{subject to} \quad c(x) \geq 0 \end{aligned} \tag{1.4.1}$$

(with $c \in \Re^m$) might be transformed into the problem

$$\underset{x \in \Re^n}{\text{minimize}} \quad B(x,\mu) \equiv F(x) + \mu \sum_{i=1}^{m} \frac{1}{c_i(x)}, \tag{1.4.2}$$

where $\mu > 0$ is a *barrier parameter* that weights the effect of the perturbation to the original function. The general idea of the classical barrier approach was to transform the original constrained problem into a *sequence* of unconstrained problems whose successive minimizers converge to the desired solution. Note that barrier methods are applicable only to problems for which a *strictly feasible point* exists, i.e., $c(x) > 0$.

In general, minimizers of $F(x)$ will be infeasible, or $F(x)$ might be unbounded below. In order to enforce feasibility of successive iterates, a modified function is introduced that resembles the original objective function but has an additional term that sharply approaches infinity at each of the constraints. That is, a *barrier* is created at the boundary of the feasible region. If the feasible region is compact there must exist a feasible (with respect to the constraints of the original problem) minimizer of the modified function. If the weight assigned to the barrier term is decreased toward zero, there exists a sequence of minimizers that constitutes a strictly feasible sequence of approximations to a constrained minimizer of the original problem. It can be appreciated from the example (1.4.2) that the smaller the value of μ the closer $B(x,\mu)$ approximates $F(x)$. However, no matter how small μ, very close to the boundary of the feasible region the two functions will differ.

If the initial estimate of the minimizer of (1.4.2) is feasible with respect to the constraints $c(x) \geq 0$, then an algorithm to solve (1.4.2) can usually be adjusted to generate a strictly feasible sequence of estimates.

Figures 1.3 and 1.4 illustrate the effect of applying a barrier transformation to a problem with one nonlinear constraint. Figure 1.3 shows the contours of the function $F(x) = x_1 x_2^2$ and the contour line corresponding to a zero value of the constraint $c(x) \equiv 2 - x_1^2 - x_2^2 \geq 0$. In Figure 1.4 no contours of the barrier function have been drawn in the infeabile region, since the function is not meant to be evaluated in that region.

The two most popular barrier functions are the *logarithmic* barrier function attributed to Frisch [Fri55], and the *inverse* barrier function (used in (1.4.2)) introduced by Carroll [Car59,Car61]. Here we consider only the logarithmic barrier function. Let x^* be a minimizer of (1.4.1), and let $x^*(\mu)$ be the minimizer of $B(x,\mu)$ in (1.4.2) for a given value of μ. Fiacco and McCormick [FM68] have shown, for a wide class of barrier functions and under quite general conditions on $F(x)$ and $c(x)$, that there exists a compact set containing x^* within which the sequence $\{x^*(\mu)\}$ converges to x^* as $\mu \to 0$.

In the 1960's, the popularity of these methods grew. The attractiveness was due in part to the sophistication of algorithms then known for unconstrained problems

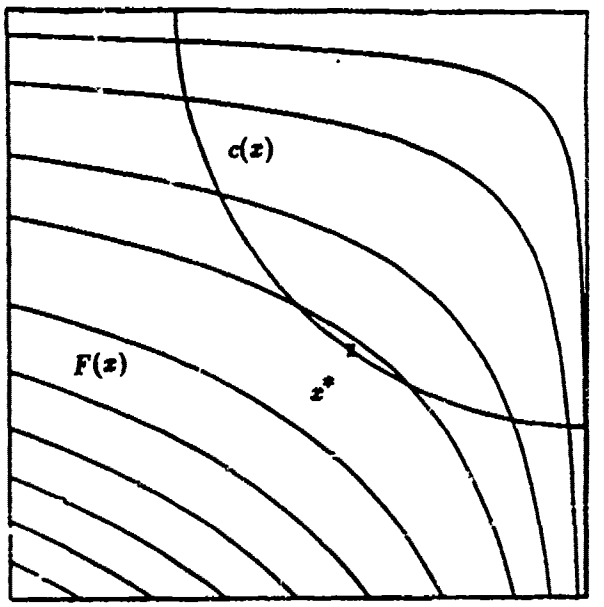

Figure 1.3: A Nonlinear Problem

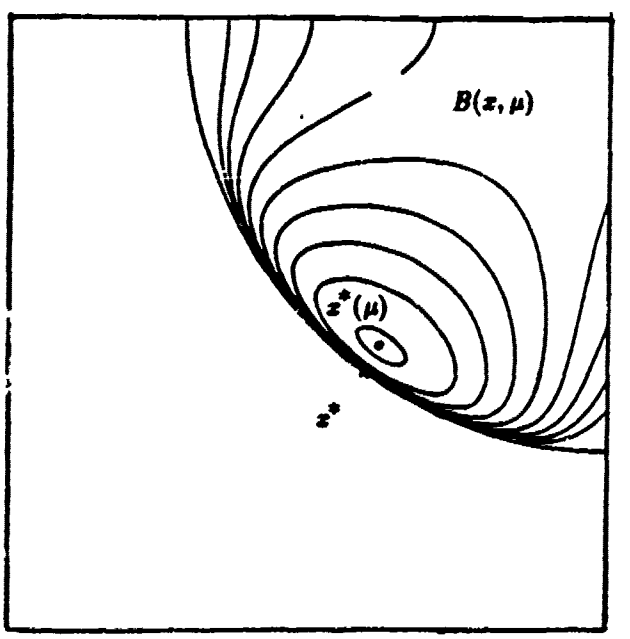

Figure 1.4: Barrier Transformation Applied to a Nonlinear Problem

compared to those known for constrained problems. Once the problems were transformed to become unconstrained, "off-the-shelf" software could (in theory) be used for their solution.

Since techniques for *nonlinear* constraints were not well understood at the time, their removal was specially advantageous. Another desirable feature of the barrier transformation is that it couples the original objective function and the inequality constraints in a manner that eliminates motion along the boundary. Hence, no special techniques are needed to move along the boundary.[1] However, solution of the barrier subproblems proved difficult and in the 1970's interest in barrier methods declined. One reason for the difficulty is that the Hessian at the solution becomes progressively more ill-conditioned as $\mu \to 0$ [Mur71b]. Most methods for unconstrained problems perform poorly on ill-conditioned problems. For example, Newton's method has a quadratic rate of convergence only if the Hessian is nonsingular (otherwise the rate is just linear). Thus, although a constrained problem has been reduced to a sequence of unconstrained problems, the latter are of the type that are difficult to solve. We should emphasize that the ill-conditioning is a feature of the barrier transformation and is independent of the condition of the original optimization problem. For an approach that circumvents this ill-conditioning, see [Wri76].

Interest in barrier methods has been revived recently, following the discovery that the much-publicized Karmarkar algorithm for linear programming is in fact equivalent to a particular type of barrier method [Kar84,GMS*86]. It is now clear that the removal of inequality constraints can be advantageous even for linear programming, since it avoids the combinatorial element of finding the active set.

It turns out that linear programming is a special case and that for such problems the usual ill-conditioning associated with barrier functions is not present (assuming the problem is not dual degenerate). The reason is that in LP's the number of constraints active at the solution is usually at least equal to the number of variables. In general, this property does not hold for QP and other nonlinear problems. At first sight, therefore, the extension of the new approaches to LP to nonlinear problems does not look promising. However, we have been able to show that provided the barrier transformation is applied in a certain manner, the inevitable ill-conditioning is benign.

The crucial element is to apply the barrier transformation to simple bounds only, instead of to general inequalities. For example, constraints of the form $Ax \geq b$ are replaced by the equivalent constraints $Ax - s = b$ and $s \geq 0$. The bounds $s \geq 0$ are then eliminated by the barrier transformation. The barrier subproblems now have linear equality constraints. In general this is not a drawback since many problems will already have some naturally occurring equality constraints.

[1] Motion along the boundary is undesirably complex when the constraint surface is nonlinear.

1.5 Large-scale Quadratic Programming

While general algorithms for quadratic programming are well understood, computationally attractive algorithms for large-scale quadratic programming are scarce. The definition of a large problem naturally depends on the size of the machine being used. It is generally agreed that large problems involve a few thousand variables and constraints. When problems of this type arise there is often considerable structure to both the Hessian and constraint matrices. In particular, the matrices typically have an average of only 5 to 10 nonzero elements per column [Lus89]. Enormous savings in computational effort as well as storage can be achieved if the QP algorithm takes advantage of such structure.

Within the last 15 years a few algorithms have appeared for large-scale quadratic programming. These include Tomlin's sparse implementation of Lemke's method [Tom76b,Tom76c,Tom76a], and Gould's method [Gou86a]. The latter is an active-set method that uses a sparse solver for matrices derived from KKT systems.

Among the best known algorithms for large-scale optimization is the MINOS system due to Murtagh and Saunders [MS83]. While designed for general nonlinear programming, MINOS can be applied to quadratic programs. For linearly constrained problems, an active-set method is used. A characteristic is that advantage is taken of sparsity in the constraint matrix but not in the Hessian. Instead, MINOS works with a dense approximation to the so-called *reduced* Hessian, and relies on the fact that for many problems this matrix is relatively small. If the reduced Hessian is of low dimension (say less than 200), the algorithm is quite efficient. The effectiveness of the method decreases as the size of the reduced Hessian increases. We defer a more complete description to Chapter 8, where the performance of the barrier method is compared with MINOS.

1.6 Complexity Results

For an introduction to computational complexity, the theory of NP-completeness and its significance within this context, techniques for analyzing the complexity of algorithms, and practical approaches to some intractable problems, see [GJ79,PS82]. For a study of the computational complexity of fundamental processes in numerical computation, see [Kar74].

Barrier methods belong to the class of *interior-point methods*, in which inequality constraints are strictly satisfied throughout.

A key theoretical property of certain interior-point methods for LP and convex QP is that they are *polynomial-time algorithms*; i.e., the number of iterations required to reach a specified accuracy in the solution is a polynomial function of n, the number of variables.

For example, for linear programs and strictly convex QP's, the duality gap $\gamma = x^T z$ is one measure of distance from a solution, assuming x and z satisfy the primal and dual constraints respectively. For a *primal-dual* algorithm that maintains such points (x, z), a typical complexity result might state the following. For a given accuracy parameter δ, if $\gamma_0 = x_0^T z_0$ is the duality gap for some initial primal and dual feasible point (x_0, z_0), the number of iterations required to reduce the duality gap to a specified value $\gamma > 0$ is at most N. where N satisfies a bound of the form

$$N < C\sqrt{n}\log(\gamma_0/\delta),$$

for a moderate constant C.

Further complexity results are often given in terms of L, the "length" or "size" of the data specifying the problem. Typically,

$$L = \sum \lceil \log_2(|d_i| + 1) \rceil,$$

where d_i ranges over all nonzero elements of A, b, c and H. In other words, L is the number of bits of data needed to specify the problem. In many cases it is assumed that the data elements d_i are rational.

If the accuracy is required to be $\gamma \leq 2^{-2L}$, it may be possible to state for some algorithm that in the optimal solution (x^*, z^*),

$$x_j^* = 0 \text{ if } x_j \leq 2^{-L} \quad \text{and} \quad z_j^* = 0 \text{ if } z_j \leq 2^{-L}.$$

The remaining components of the solution can be computed in $O(n^3)$ operations. Thus, the condition

$$x^T z \leq 2^{-2L}$$

could theoretically be used as a stopping criterion for the algorithm.

Karmarkar [Kar84] was the first to prove polynomiality for an interior-point method for LP. Many such algorithms have since been developed for LP, convex QP, and linear complementarity problems. They are variously known as

- potential-reduction algorithms,
- path-following algorithms,
- barrier algorithms,
- affine-scaling algorithms,
- projective-scaling algorithms.

It should be emphasized that all these algorithms are closely related despite the many different names.

The main LP/QP algorithm developed and implemented here is of the barrier type. For the special case of LP, polynomiality was proved by Gonzaga [Gon87] and by Roos and Vial [RV88]. In the latter case, the barrier parameter μ is reduced occasionally by an arbitrary constant factor in the range $(0, 1)$. The algorithms presented in this thesis are similar in spirit to this approach.

A great deal of background and unifying theory has recently been given by Kojima *et al.* [KMNY90] on interior-point algorithms for linear complementarity problems, including LP and convex QP. Their results apply to primal-dual algorithms such as those described in Chapter 6.

Further complexity results for convex QP algorithms have been given by Ye [Ye87] and by Ye and Tse [YT86].

While proofs of polynomiality are of considerable theoretical interest, the theoretical bounds that have been obtained are enormously greater than the actual numbers of iterations that are typically achieved in practice. If this were not the case, barrier methods would be of little practical value. Some steps towards bounding the *expected* number of iterations have been taken recently by Todd *et al.* [TMY90], but again for the LP case only.

1.7 Thesis Contents

Elsewhere, the principal form of research on interior-point methods for quadratic programming has been to investigate their complexity properties (for example, see Kojima *et al.* [KMNY90]). In this thesis the focus is on defining *practical* algorithms and facing the many numerical issues that arise (*cf.* [Meh89,Meh90,LMS89,LMS90, CLMS90]). A key question that is not addressed in the basic theory of barrier functions is how to determine (or approximate) the solution of the subproblem. Much of the thesis is directed at answering this question.

Although our prime interest is in solving large-scale QP's, much of the work presented is equally applicable to large-scale optimization problems whose objective function is an arbitrary nonlinear function that is twice continuously differentiable.

In Chapter 2 we state the necessary and sufficient conditions for a solution of a general quadratic program. For reference purposes we present several ways of formulating a QP and introduce the concept of duality within the quadratic programming context. We briefly describe active-set methods, one of the most popular class of methods for quadratic programming, we motivate the interest in new approaches, such as barrier methods, by discussing some of the complications that arise in developing active-set methods for large-scale QP's. Specifically, we show the need to circumvent the combinatorial aspect present in such methods.

In Chapter 3 we review the fundamental properties of barrier functions and the classical theory of barrier methods for the solution of a general nonlinear problem with inequality contraints (NIP). Optimality conditions for a minimizer of NIP are given, and we discuss the relationship of minimizers of the barrier subproblems to those of NIP. A model barrier algorithm is presented for the solution of nonlinear programs subject to a mixture of linear equality and general inequality constraints.

In Chapter 4 we discuss algorithms for finding or approximating minimizers of the barrier subproblem (the minimization of a nonlinear function subject to linear equality constraints). It will be seen that for the large-scale case, such algorithms are both rare and complex. Optimality conditions for the solution of the barrier subproblem are given.

Regardless of the method used to solve the subproblem, there are some special difficulties not present in the usual equality-constrained optimization problem. For example, with a normal problem it is trivial to provide an initial point that satisfies the linear equality constraints, but here the point must also be strictly interior with respect to the bounds. We propose an alternative to a general-purpose algorithm that takes specific advantage of the form of the barrier subproblem.

We are also interested in exactly *how* the sequence may be computed efficiently. A key issue in algorithms for large-scale problems is how best to solve the linear systems that define the iterative process. A feature of barrier algorithms for LP is that it is possible to define the sequence by solving symmetric positive definite systems. As it happens, algorithms to solve such systems are highly developed. In the QP case the systems are symmetric but indefinite. Such systems are inherently more difficult to solve (in part because numerical stability depends upon choosing a suitable ordering of the equations). We describe procedures to solve such systems.

Our interest is not simply in defining an iterative sequence that converges to a QP solution but also in showing that when the sequence is computed in finite precision it converges to the solution of a neighboring problem.

To prove that the proposed barrier algorithm is numerically viable, we must also show that the solution to the subproblems can be computed in a stable fashion. To this end, Chapter 5 presents a detailed sensitivity analysis of the KKT system arising from barrier methods. A key result of this analysis is that the ill-conditioning normally associated with the Hessian in barrier methods is benign when the logarithmic barrier function is applied to a particular formulation of a quadratic program. When only simple bounds are enforced by the barrier functions, the subproblems can be solved to the required accuracy in spite of the severe ill-conditioning. Thus, a stable barrier algorithm is realized.

In Chapter 6 we study some special cases of quadratic programs and the design of practical methods that take advantage of the special properties of such problems. Specifically we consider the following special cases: when the only constraints present

are simple bound constraints, when the Hessian is trivially invertible, and when the reduced Hessian is positive semidefinite. The latter case exhibits the key property that the minimizer of the barrier subproblem is unique. For such QP's it is possible to develop primal, dual and primal-dual methods. The convergence of these special methods is discussed. In the linear programming context, affine methods have been proposed. We investigate whether similar approaches are possible for solving convex quadratic programs.

For linear programming problems, there exists an input/output format that has become the industry standard and is recognized by all commercial mathematical programming systems—the so-called Mathematical Program System (MPS) format. Unfortunately, there is no equivalent standard format for quadratic programming. In Chapter 7 we define a standard for specifying a QP that we have termed the QPS format. The basis of designing the QPS format has been to adhere as closely as possible to the desirable features present in the MPS format. A feature of the new format is that it is simple to convert an MPS file into a QPS file. (A program to do this has been written and used in our computational experiments.)

An implementation of the barrier QP algorithm (termed BARQP) has been developed. In Chapter 8 the details of the implementation are described. Barrier algorithms are sensitive to many of the parameters used in their definition. A poor choice of such parameters can lead to very inefficient algorithms. Part of the purpose of the numerical experimentation has been to determine a good choice for these critical parameters.

In order to extend the usefulness of the implementation to as wide a class of problems as possible, attention is given to a number of numerical issues. For example, it may be that the constraint matrix is ill-conditioned. Inevitably such ill-conditioning is reflected in the KKT system. If the ill-conditioning is sufficiently severe the algorithms to solve such systems break down. We show how to modify the systems to ensure that the ill-conditioning does not exceed the bound for which the algorithms break down.

In contrast to the LP case, there does not exist a standard set of QP test problems. Most of the problems we have used in our tests have been derived from a well known LP test set by changing the objective function to be quadratic. We have also tested our algorithm to see how effective it is at solving LP's. We should stress that the implementation is still in a relatively primitive form. It has not been our purpose to develop an implementation suitable for general distribution. Results are presented and a comparison made with the performance of MINOS (a well established package for the solution of nonlinear optimization problems). These results are not meant as an accurate reflection of the performance of a barrier QP algorithm but are intended to serve as an indication of the algorithm's potential.

Chapter 2

Fundamentals of Quadratic Programming

In this chapter we present the fundamental properties and characteristics of the general quadratic programming (QP) problem. The optimality conditions of the problem are given, and additionally we motivate some of the issues that arise in developing practical methods for solving large-scale quadratic programs. We present a brief description of active-set methods for dense QP problems and highlight the difficulties in extending these algorithms to the solution of large-scale problems. The latter is intended to explain the motivation behind new approaches, such as barrier methods.

2.1 Equality-Constrained Quadratic Programming

We first consider the equality constrained quadratic problem (EQP). This is the simplest category of QP problems, yet it is of fundamental importance in the theory of nonlinear programming. Problems of the form EQP occur as subproblems within many active-set methods for inequality constrained quadratic programs as well as in sequential quadratic programming methods for solving general nonlinear programs.

Formally, the EQP problem can be expressed as

$$\text{EQP} \qquad \begin{array}{ll} \underset{x \in \Re^n}{\text{minimize}} & \varphi(x) \equiv c^T x + \frac{1}{2} x^T H x \\ \text{subject to} & Ax = b, \end{array} \tag{2.1.1}$$

where c is an n-vector, b is an m-vector, A is an $m \times n$ matrix of rank m, and H is an $n \times n$ symmetric matrix. The gradient of φ is the linear function $g(x) = c + Hx$. Note that H is the Hessian matrix (of second partial derivatives) of the quadratic objective function.

When a set of m linear constraints is imposed on an n-dimensional minimization problem, intuitively we expect to reduce the dimensionality of the optimization to $n - m$. Formally, the constraints $Ax = b$ define two complementary subspaces: the m-dimensional subspace defined by the rows of A and the complementary subspace of vectors orthogonal to the rows of A. Let Y denote any matrix whose columns form a basis for the *range space* of A^T, and Z denote a matrix whose columns form a basis for the *null space* of A, i.e., $AZ = 0$. We emphasize that the matrices Y and Z are not unique.

Any vector x can be written as a linear combination of the columns of Y and Z, namely

$$x = Yx_Y + Zx_Z$$

for some m-vector x_Y and an $(n-m)$-vector x_Z. If x is a feasible point of EQP, the vector x_Y is the solution of the linear system

$$AYx_Y = b. \tag{2.1.2}$$

When A has full row rank, the matrix AY is nonsingular and the vector x_Y is unique. Hence, the constraints entirely determine the range-space portion of the solution x^* of EQP; that is, $x_Y = x_Y^*$. This is precisely the expected reduction in dimensionality. It is convenient both conceptually and computationally to express any feasible point x of EQP as

$$x = Yx_Y^* + Zx_Z,$$

since it implies that solving the constrained problem EQP can be viewed as minimizing the following *unconstrained* problem in the variables x_Z:

$$\underset{x_Z \in \Re^{n-m}}{\text{minimize}} \quad \bar{\varphi}(x_Z) \equiv (c + HYx_Y^*)^T Zx_Z + \tfrac{1}{2} x_Z^T (Z^T HZ) x_Z. \tag{2.1.3}$$

The matrix Z^THZ is known as the *reduced Hessian.* The following lemma, which summarizes the characterization of a minimizer of EQP for the case when A has full row rank, follows immediately from the optimality conditions of problem (2.1.3).

LEMMA 2.1.1. *Let A be an $m \times n$ matrix of rank m, let Z denote a basis for the null space of A, and let $x = Yx_Y^* + Zx_Z$, where $AYx_Y^* = b$ and Y is a basis for the range space of A^T. Then*

(i) *EQP has a strong local minimizer at x^* if and only if x_Z^* is a stationary point of $\bar{\varphi}(x_Z)$ and Z^THZ is positive definite;*

(ii) *EQP has an infinite number of weak solutions if and only if there is a stationary point of $\bar{\varphi}(x_Z)$ and Z^THZ is positive semidefinite and singular;*

(iii) *EQP has no solution if no stationary point of $\bar{\varphi}(x_Z)$ exists.*

Note that when the local minimizer of EQP exists, it is also a global minimizer.

If A is not of full row rank, in addition to the above conditions it is necessary for the vector b to lie in the column space of A. In theory we can always remove dependent constraints such that the resulting A has full row rank; however there may be numerical difficulties in recognizing this situation. In general the cost of achieving full rank is not negligible.

The first-order necessary conditions state that a solution x^* of EQP must satisfy

$$\nabla\bar{\varphi}(x_Z^*) = Z^T\nabla\varphi(x^*) = Z^Tg(x^*) = Z^T(c + Hx^*) = 0, \tag{2.1.4}$$

or equivalently, there exists π^* such that $Ax^* = b$ and

$$g(x^*) = c + Hx^* = A^T\pi^* \tag{2.1.5}$$

for some vector of Lagrange multipliers π^*. Thus, x^* and π^* satisfy the equation

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} -x^* \\ \pi^* \end{pmatrix} = \begin{pmatrix} c \\ -b \end{pmatrix}. \tag{2.1.6}$$

The matrix

$$K = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}$$

is referred to as the Karush-Kuhn-Tucker (KKT) matrix, and (2.1.6) is the KKT system of equations. In Chapter 3 we discuss the KKT system in the context of the barrier method. The above system can also be derived by considering the *Lagrangian function* associated with the EQP problem, namely

$$L(x,\pi) \equiv c^Tx + \tfrac{1}{2}x^THx - \pi^T(Ax - b).$$

It can be easily verified that if x^* and π^* satisfy (2.1.6), then (x^*,π^*) is a *stationary point* of the Lagrangian function.

Given any point (x,π), let (p,q) be the step to the solution (x^*,π^*), so that $x^* = x + p$ and $\pi^* = \pi + q$. It is convenient to rewrite equations (2.1.6) in terms of p and q:

$$K\begin{pmatrix} -p \\ q \end{pmatrix} = \begin{pmatrix} g_L \\ r \end{pmatrix}, \tag{2.1.7}$$

where $g_L = c + Hx - A^T\pi$ is the gradient of the Lagrangian with respect to x, and $r = Ax - b$ is the residual for the constraints.

We can also characterize a minimizer of EQP in terms of the *inertia* of the KKT matrix.

DEFINITION 2.1.1. *The inertia of a symmetric matrix K is the triplet*

$$In(K) = (i_+, i_-, i_0),$$

where i_+, i_- and i_0 are the number of positive, negative and zero eigenvalues of K respectively.

The following lemma gives an important relationship between the KKT matrix and the reduced Hessian Z^THZ.

LEMMA 2.1.2. *Let H be an $n \times n$ symmetric matrix and A an $m \times n$ matrix of full row rank. If Z is a basis for the null space of A, then*

$$In(K) = In\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} = In(Z^THZ) + (m, m, 0). \tag{2.1.8}$$

Proof. For a proof, see [Gou85].

Notice that the KKT matrix is nonsingular if and only if the reduced Hessian is nonsingular. Using (2.1.8), Lemma 2.1.1 can also be stated as follows.

LEMMA 2.1.3. *Let A be an $m \times n$ matrix of rank m, let Z denote a basis for the null space of A, and let $x = Yx_Y^* + Zx_Z$, where $AYx_Y^* = b$ and Y is a basis for the range space of A^T. Then*

(i) *EQP has a strong local minimizer at x^* if x_Z^* is a stationary point of EQP and $In(K) = (n, m, 0)$;*

(ii) *EQP has an infinite number of weak solutions if there is a stationary point of EQP and $In(K) = (n - t, m, t)$, with $t > 0$;*

(iii) *EQP has no solution if no stationary point of EQP exists.*

A detailed discussion of the conditions for the existence and uniqueness of solutions of the EQP problem can be found in [Gou85].

2.2 Inequality-Constrained Quadratic Programming

The *general quadratic programming problem* is to find a local minimizer of a quadratic function subject to linear constraints. There are many mathematically equivalent formulations of a quadratic program (although not all have the same computational behavior). We first consider the problem in the following pure inequality form:

$$\text{IQP} \qquad \begin{array}{ll} \underset{x \in \Re^n}{\text{minimize}} & \varphi(x) \equiv c^Tx + \frac{1}{2}x^THx \\ \text{subject to} & \mathcal{A}x \geq \beta, \end{array} \tag{2.2.1}$$

where the Hessian H is an $n \times n$ symmetric matrix, $\mathcal{A}$ is an $m \times n$ matrix, and c and β are vectors of dimension n and m respectively. An important special case of the above problem occurs when H is positive definite.

We give the first-order and second-order necessary conditions for a solution of IQP. In order to do so, it is important to distinguish between constraints that are satisfied exactly at the solution, and those that are not binding.

DEFINITION 2.2.1. *The point $\hat{x}$ is said to be feasible with respect to the inequality constraint $a_i^T x \geq \beta_i$ if $a_i^T \hat{x} \geq \beta_i$. (That is, the constraint is satisfied at $\hat{x}$.) The constraint $a_i^T x \geq \beta_i$ is said to be active at $\hat{x}$ if $a_i^T \hat{x} = \beta_i$ and inactive if $a_i^T \hat{x} > \beta_i$. If $a_i^T \hat{x} < \beta_i$, $\hat{x}$ is infeasible, and the constraint is said to be violated at $\hat{x}$.*

The active constraints play a special role because they restrict feasible perturbations. If a constraint is inactive at the point x, then it will remain inactive for *any* perturbation in a sufficiently small neighborhood. However, an *active* constraint may be violated by certain perturbations.

LEMMA 2.2.1. *(Necessary conditions.) Let A be the $t \times n$ matrix containing only those rows of A for which the corresponding constraint is active at the point x^*, let b be the corresponding elements of β, and let Z be a basis for the null space of A. The point x^* is a minimizer of IQP only if there exists a vector π^* such that*

(i) $A^T\pi^* = Hx^* + c$, *or equivalently* $Z^T(Hx^* + c) = 0$;

(ii) $Ax^* \geq \beta$; $Ax^* = b$;

(iii) $\pi_i^* \geq 0$, $i = 1, \ldots, t$; *and*

(iv) Z^THZ *is positive semidefinite.*

The conditions (i), (ii) and (iii) are referred to as the first-order necessary conditions and (iv) is a second-order necessary condition. In contrast to the EQP case, the optimality conditions for IQP restrict the sign of the Lagrange multipliers associated with the active constraints at the solution.

LEMMA 2.2.2. *Sufficient conditions for x^* to be a solution of IQP are given by*

a) $A^T\pi^* = Hx^* + c$, *or equivalently* $Z^T(Hx^* + c) = 0$;

b) $Ax^* \geq \beta$; $Ax^* = b$;

c) $\pi_i^* > 0$, $i = 1, \ldots, t$; *and*

d) Z^THZ *is positive definite.*

Under such circumstances, x^* is an isolated minimizer.

An extra complication over the equality-constrained case arises when the Lagrange multiplier corresponding to an active constraint is zero. In this case, any set of sufficient conditions must include certain restrictions to account for perturbations that are binding for all the constraints with positive Lagrange multipliers but that may be binding or non-binding for constraints with zero Lagrange multipliers.

When the sufficient conditions hold, x^* is not only optimal but is also *locally unique*, i.e., $\varphi(x^*) < \varphi(x)$ for all feasible x in a neighborhood of x^* ($x^* \neq x$). The gap between the necessary conditions and the sufficient conditions arises from the possibility of one or more zero Lagrange multipliers and/or a positive definite and singular reduced Hessian. Sets of conditions that are simultaneously necessary *and* sufficient are surprisingly complicated and they can seldom be verified in practice.

When the necessary conditions are satisfied but the sufficient conditions are not, x^* may or may not be a local solution of (2.2.1)—for example, a feasible direction of decrease may exist. Verification of optimality in such instances requires further information, and is in general an NP-hard problem (see Murty and Kabadi [MK87] and Pardalos and Schnitger [PS88]) that is equivalent to the *copositivity problem* of quadratic programming (see, e.g., Contesse [Con80] and Majthay [Maj71]).

2.3 Convex and Nonconvex Quadratic Programming

Quadratic programs can be grouped into two broad categories: convex and nonconvex QP. The former category exhibits some interesting properties; for example, a local minimizer of a convex QP is also a global minimizer. This section is intended to point out briefly why computing the solution of a nonconvex quadratic program constitutes a much more difficult task than that of solving a convex quadratic program.

Consider an iterative procedure for computing the solution of a quadratic program. Let $Z_k^T H Z_k$ be the reduced Hessian matrix at the point x_k. When $Z_k^T H Z_k$ is positive definite, the search direction p_k at x_k can be obtained by solving the Newton equations

$$Z_k^T H Z_k p_z = -Z_k^T g_k \tag{2.3.1}$$

and setting $p_k = Z p_z$. It can be verified that p_k is a descent direction for the quadratic function. Furthermore, the minimizer in the subspace defined by Z_k is unique.

A *strictly convex quadratic program* is one for which H is positive definite. In this case, $Z_k^T H Z_k$ is known to be positive definite at every iterate x_k, and hence p_k is always well defined. For *convex* QP's, $Z_k^T H Z_k$ might be only positive semidefinite. However, every minimizer is a global minimizer.

In nonconvex quadratic programs, complications arise when the reduced Hessian $Z_k^T H Z_k$ is indefinite. In this case, a point satisfying the first-order necessary conditions

for optimality is not a local minimizer in the current null space. Indeed there may be no point on a given set of constraints that satisfies the first-order necessary conditions. Also the direction defined by (2.3.1) is not necessarily a descent direction for the quadratic function. Hence, a mechanism for computing a feasible direction of descent must be provided. In terms of efficiency, it is desirable for such a mechanism to retain as much information about the present (indefinite) Hessian as possible. Hence, standard approaches for modifying an indefinite matrix may not be suitable. Care must be exercised in updating a factorization of an indefinite matrix, since there is a danger of numerical instability (see Gill *et al.* [GMSW84c] where the Hessian is allowed to have *one* negative eigenvalue).

Finally, with nonconvex quadratic programs there may exist so-called *dead points* at which it is very difficult to verify optimality. At such points, all conventional quadratic programming methods will find it difficult to proceed, since it can be shown that the problem of distinguishing a dead point that is not a minimizer is an NP-hard problem (see Forsgren, Gill and Murray [FGM89b] for a precise definition of a dead point and a computational scheme within the context of inertia-controlling methods for QP that will attempt to determine if a dead point is a local minimizer). We emphasize that this difficulty is inherent in the problem, and is independent of the solution method.

2.4 Formulating a QP Problem

The quadratic programming problem can be expressed in several mathematically equivalent forms. For reference purposes, we introduce in this section several formulations.

The General Form.

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & c^Tx + \frac{1}{2}x^THx \\ \text{subject to} & A_1x = b_1 \\ & A_2x \geq b_2 \\ & \ell \leq x \leq u, \end{array} \tag{2.4.1}$$

where A_1 and A_2 are of dimension $m_1 \times n$ and $m_2 \times n$ respectively. Constraints $A_1x = b_1$ and $A_2x \geq b_2$ are called the *general* constraints. Thus, formulation (2.4.1) includes a mixture of general equality and inequality constraints as well as simple upper and lower bounds on the variables.

The Standard Form.

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & c^Tx + \frac{1}{2}x^THx \\ \text{subject to} & Ax = b \\ & \ell \leq x \leq u, \end{array} \tag{2.4.2}$$

where all the general constraints are equalities, and the only inequalities are upper and lower bounds on the variables. For expository purposes we shall usually omit the upper bounds and take $\ell = 0$.

Any quadratic programming problem can be converted to *standard form*. It is sometimes convenient to transform a given QP problem into this form. For instance, for large-scale quadratic programs, it can be algorithmically advantageous to assume that the constraints are posed as in (2.4.2) (see, for example, Gill *et al.* [GMSW87, GMSW88]). Indeed it will be seen later that rather than being convenient, it is essential to the success of the barrier methods described here to convert the problem to standard form.

A QP problem stated in the form (2.4.1) can be converted into the standard form by introducing slack variables or by using duality. We shall illustrate the latter in Section 2.5.

In particular, a general inequality constraint $a_i^Tx \geq \beta_i$ can be replaced by the equality constraint $a_i^Tx - s_i = \beta_i$, and the standard-form version of the problem includes an additional *slack* variable subject to the bound $s_i \geq 0$. Slack variables have many special features; for instance, they do not appear in the objective function. By adding m_2 slack variables s, problem (2.4.1) can be transformed into standard form as follows:

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n,\ s\in\mathbb{R}^{m_2}}{\text{minimize}} & c^Tx + \frac{1}{2}x^THx \\ \text{subject to} & A_1x \quad = b_1 \\ & A_2x - s = b_2 \\ & \ell \leq x \leq u \\ & 0 \leq s \leq \infty. \end{array} \tag{2.4.3}$$

2.5 Duality in Quadratic Programming

The optimality conditions for linear and quadratic programming problems involve not only the variables x associated with each column of the constraint matrix but also Lagrange multipliers associated with each row. An interesting theory (*duality theory*) has been developed to explore the relationships between these two sets of variables. Traditionally, the variables x of the "original" problem are called the *primal* variables, and the original problem is denoted the *primal*. The Lagrange multipliers are sometimes referred to as *dual* variables.

The appearance of duality theory in linear programming goes back to the classical minimax theorem of von Neumann and was first explicitly given by Gale, Kuhn and Tucker (see [GKT51]). The concept of duality also applies to quadratic programs. Duality in nonlinear programming started with the results on quadratic programming given by Dennis [Den57,Den59].

We shall state some of the relationships between certain primal and dual QP programs. In what follows we assume that H is positive definite. Duality results can be extended to more general H (see Murray [Mur69]), but the conditions are somewhat complicated and difficult to verify in practice.

Primal QP (standard form):

$$\begin{aligned} \min_{x\in\Re^n} \quad & f_p = c^T x + \tfrac{1}{2} x^T H x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned} \tag{2.5.1}$$

Dual QP:

$$\begin{aligned} \max_{y\in\Re^m,\ w\in\Re^n} \quad & f_d = b^T y - \tfrac{1}{2} w^T H w \\ \text{subject to} \quad & A^T y - Hw \leq c. \end{aligned} \tag{2.5.2}$$

The dual (2.5.2) can itself be converted to standard form by adding n nonnegative slack variables z, thereby giving the following standard-form dual:

$$\begin{aligned} \max_{y\in\Re^m,\ w,z\in\Re^n} \quad & f_d = b^T y - \tfrac{1}{2} w^T H w \\ \text{subject to} \quad & A^T y - Hw + z = c, \quad z \geq 0. \end{aligned} \tag{2.5.3}$$

If the original problem is in pure inequality form we may also give a dual form:

Primal QP (inequality contraints):

$$\begin{aligned} \min_{x\in\Re^n} \quad & f_p = c^T x + \tfrac{1}{2} x^T H x \\ \text{subject to} \quad & \mathcal{A}x \geq \beta. \end{aligned} \tag{2.5.4}$$

Dual QP:

$$\begin{aligned} \min_{y\in\Re^m,\ \ x\in\Re^n} \quad & f_d = \beta^T y - \tfrac{1}{2} w^T H w \\ \text{subject to} \quad & \mathcal{A}^T y - Hw = c, \quad y \geq 0. \end{aligned} \tag{2.5.5}$$

The standard and inequality forms are included in the following general form:

Primal QP:

$$\min_{x,y} \quad f_p = c^T x + \tfrac{1}{2} x^T H x + a^T y + \tfrac{1}{2} y^T G y$$

$$\text{subject to} \quad \begin{aligned} Ax + Dy &\geq b \\ Bx + Ey &= d \\ x &\geq 0. \end{aligned}$$

Dual QP:

$$\max_{u,v,w,y} \quad f_d = b^T u + d^T v - \tfrac{1}{2} w^T H w - \tfrac{1}{2} y^T G y$$

$$\text{subject to} \quad \begin{aligned} A^T u + B^T v &\leq c + Hw \\ D^T u + E^T v &= a + Gy \\ u &\geq 0. \end{aligned}$$

The key relationship between the primal and dual is that from the solution of one problem we may recover the solution of the other. For example, in the case of the primal in standard form let x^* denote the minimizer, π^* the Lagrange multipliers of the equality constraints, and z^* the Lagrange multipliers of the bound constraints. It follows that the solution of the dual problem is given by $y = \pi^*$, $w = x^*$ and $z = z^*$. Obviously if the dual problem were solved we could recover the solution of the primal problem.

The relationship between the two solutions may be verified by observing that the necessary conditions for a solution of the two problems reduce to the same system of equations. Another interesting relationship between the primal and dual problems is that for any primal feasible point and any dual feasible point we have

$$f_p - f_d \geq 0.$$

To show why this result is true we prove it when the primal is in standard form. If x is primal feasible then

$$Ax = b, \qquad x \geq 0. \tag{2.5.6}$$

If (w, y, z) is dual feasible then

$$A^T y - Hw + z = c. \tag{2.5.7}$$

Premultiplying (2.5.7) by x^T and substituting b for Ax we obtain

$$\begin{aligned} c^T x &= x^T A^T y - x^T H w + x^T z \\ &= b^T y - x^T H w + x^T z. \end{aligned}$$

Since

$$(x-w)^T H(x-w) = x^T Hx - 2x^T Hw + w^T Hw,$$

we get

$$c^T x + \tfrac{1}{2}x^T Hx = b^T y - \tfrac{1}{2}w^T Hw + \tfrac{1}{2}(x-w)^T H(x-w) + x^T z.$$

Hence,

$$f_p(x) - f_d(w,y,z) = x^T z + \tfrac{1}{2}(x-w)^T H(x-w) \geq 0. \tag{2.5.8}$$

If a primal feasible point and a dual feasible point were known then an upper and lower bound on the solution would also be known. The difference $f_p - f_d$ is referred to as the duality gap. It is easily seen from the above result that

$$f_p(x^*) - f_d(x^*, y^*, z^*) = 0. \tag{2.5.9}$$

2.6 Active-set Methods

We briefly describe one of the most popular classes of methods for quadratic programming, namely *active-set* methods. Our primary purpose is to illustrate the difficulties that arise in applying such methods to large-scale problems, thus motivating the barrier approach. It will also give us a basis for comparing the efficiency of barrier methods.

If the active set at the solution were known *a priori*, the solution to the IQP problem (2.2.1) could be determined by solving a single KKT system of equations. Active-set methods are iterative methods that maintain an estimate of the set of constraints active at the solution—called the *working set*—which is a linearly independent set of constraints. In the active-set methods described in this section, it is assumed that at a particular iterate x, all constraints in the working-set are active.

We shall illustrate the steps of a QP method for a *primal-feasible active-set method.* Each iteration has the following general structure: given the current iterate x, the next iterate is defined by

$$\bar{x} = x + \alpha p,$$

where the vector p is the *search direction*, and the nonnegative scalar α is the *steplength.* An initial *feasibility phase* is performed to find a point that satisfies the constraints of (2.2.1), and all iterates are thereafter constructed to retain feasibility. Thus, if A_W denotes the working-set matrix and b_W the associated right-hand side vector, we have $A_W x = b_W$.

The search direction p is defined as the solution of the following *equality-constrained* QP:

$$\begin{aligned} \underset{p \in \mathbb{R}^n}{\text{minimize}} \quad & g^T p + \tfrac{1}{2}p^T H p \\ \text{subject to} \quad & A_W p = 0, \end{aligned}$$

where g denotes $g(x)$, the gradient of φ at the current iterate. The constraints $A_W p = 0$ ensure that constraints in the working set remain unaltered by any move along p.

The following (standard) terminology is useful in characterizing the relationship between p and φ:

$$p \text{ is a } \begin{cases} \text{descent direction} & \text{if } g^T p < 0; \\ \text{direction of negative curvature} & \text{if } p^T H p < 0. \end{cases}$$

Because φ is quadratic,

$$\varphi(x + \alpha p) = \varphi(x) + \alpha g^T p + \tfrac{1}{2}\alpha^2 p^T H p. \tag{2.6.1}$$

This relation shows that every direction p for which φ decreases must be either a descent direction, or a direction of negative curvature with $g^T p = 0$. If $g^T p < 0$ and $p^T H p > 0$, we see from (2.6.1) that $\varphi(x + \alpha p) < \varphi(x)$ for all $0 < \alpha < \tau$, where $\tau = -2g^T p / p^T H p$. If $g^T p < 0$ and $p^T H p \leq 0$, or if $g^T p = 0$ and $p^T H p < 0$, (2.6.1) shows that φ is monotonically decreasing along p, i.e., $\varphi(x+\alpha p) < \varphi(x)$ for *all* $\alpha > 0$.

If the reduced Hessian with respect to the working set is positive definite, a minimizer on the subspace defined by these constraints can be found. The step p to the minimizer on the current working set satisfies the equation

$$\begin{pmatrix} H & A_W^T \\ A_W & 0 \end{pmatrix} \begin{pmatrix} p \\ -\pi_W \end{pmatrix} = - \begin{pmatrix} c + Hx \\ 0 \end{pmatrix}, \tag{2.6.2}$$

where π_W contains the Lagrange multipliers corresponding to the constraints in the working set. The minimizer on the subspace is then found as $x + p$. However, since there are other constraints present in the problem, the point $x + p$ may not be feasible. In this situation, the maximum feasible step $\alpha = \alpha_{max}$ along p is computed. If $\alpha_{max} < 1$, the constraint that becomes active at the new iterate is added to the working set. If the unit step along p is accepted, a minimizer on the subspace has been found. The Lagrange multiplier vector is then examined to see if $\pi_W \geq 0$. If not, a constraint corresponding to a negative multiplier is deleted from the working set.

An active-set method of this kind usually deletes and adds only one constraint at a time. Such a method allows use of computationally efficient updating schemes for the matrix factorizations. Strategies that add and delete more than one constraint at a time are possible.

One class of active-set methods is the *inertia-controlling methods* for quadratic programming (ICQP methods); for example see [Fle71,GMSW84c,GMSW88,Gou86a]. Such methods control the working set so that the associated reduced Hessian has

at most one nonpositive eigenvalue. (For a full description of ICQP methods see [GMSW88]). Usually only one constraint is added or deleted at a time. The working set at the initial point must be chosen so that the reduced Hessian is positive definite. By appending certain *artificial* or *temporary* constraints to the problem this condition can be satisfied at an arbitrary point. A necessary requirement for an artificial constraint is linear independence relative to constraints already in the working set. Artificial constraints do not restrict the feasible region, since they may be eliminated from the problem whenever the algorithm permits.

For the solution of moderate-sized quadratic programming problems, ICQP methods have proved to be very efficient. This is due in part to the use of sophisticated techniques for updating factorizations of dense matrices, and to the fact that the factorizations needed to compute a direction of descent may also be used to compute a direction of negative curvature when such a direction exists.

Unfortunately, the efficient application of active-set methods to large-scale problems is non-trivial. Only two large-scale ICQP methods have been proposed and both are based on a direct factorization of the KKT system. Gould [Gou86a] has proposed a method based on the use of an LU factorization together with procedures to update the LU factors. Gill *et al.* [GMSW87] have proposed a method based on an LBL^T factorization, which is for *symmetric indefinite* matrices (see Chapter 4 for a definition of this factorization). The updates are not performed on the factorization itself. The required KKT solution may be recovered by solving a system that augments the original KKT system. The solution of the augmented system is found by using the Schur complement of H in the original KKT matrix. (The Schur complement is defined in Equation (4.11.1).) A benefit of this approach is that a "black-box" LBL^T package may be used, which would not be possible if it were necessary to update the LBL^T factorization.

A key difference in solving large-scale problems is that updating results in an increasingly large data file (whereas in the dense case the elements of the factors are updated explicitly). Eventually the data file exceeds available memory or becomes large enough that it is worthwhile to discard it and restart with a new refactorization of the current KKT matrix.

Another difficulty in the large-scale case is the provision of artificial constraints. In the dense case a sophisticated procedure is known for generating artificial constraints that are in some sense ideal [GMSW84a, Section 5]. For example, they do not exacerbate the condition of the KKT system. Unfortunately, this procedure requires computing an orthogonal basis for the null space of A, which is not practical in the large-scale case. The only procedure that can be guaranteed to generate a suitable starting point in the large-scale case may generate a near vertex even when the reduced Hessian at the initial point has only a few negative eigenvalues. We may then require many iterations simply to eliminate the artificial constraints. Such difficulties

are not intractable in that special iterations may be performed to try and eliminate many constraints at a single iteration. However, the special iterations are themselves problematical.

A further difficulty with an active-set strategy is that the number of iterations usually grows with the size of the problem. However, it is known that the number of iterations can grow exponentially with the number of variables (see [Fat78]). Thus, in the application of active-set methods to large-scale quadratic problems the number of iterations is likely to be large and potentially may be astronomical.

It will be seen that none of the above difficulties arises in the barrier algorithms. However, such methods do have some unique complications of their own. It is then a question of which difficulties are the more serious when solving a specific problem.

Chapter 3

Barrier Methods

In the previous chapter we considered some of the difficulties in applying conventional methods to large-scale problems. Specifically, we emphasized the combinatorial aspect introduced by active-set methods. In this chapter we present the fundamental properties of barrier functions and give a model barrier algorithm for the solution of quadratic programs. The observed performance of barrier methods is that the required number of iterations to achieve some specified approximation to the solution is largely independent of the number of constraints and the number of variables. Although the work per iteration may be considerably greater than in active-set methods, the hope is that the *number* of iterations is so small that the total effort required by the barrier approach will be less.

3.1 Optimality Conditions

Consider a problem in which all the constraints are assumed to be inequalities:

$$\begin{array}{lll} \text{NIP} & \underset{x \in \Re^n}{\text{minimize}} & F(x) \\ & \text{subject to} & c(x) \geq 0, \end{array} \tag{3.1.1}$$

where $c(x)$ is an m-vector of nonlinear functions with i-th component $c_i(x)$, $i = 1, \ldots, m$, and F and $\{c_i\}$ are twice-continuously differentiable. Let $g(x)$ denote the gradient vector of $F(x)$, $a_i(x)$ the gradient vector of $c_i(x)$, and $\mathcal{A}(x)$ the $m \times n$ Jacobian matrix of $c(x)$. A solution of NIP will be denoted by x^*.

Let $\hat{c}(x)$ denote the vector of constraints that are *active* at x, and $A(x)$ the Jacobian of $\hat{c}$. We emphasize that $A(x)$ includes only the gradients of the *active* constraints, whereas $\mathcal{A}(x)$ is the Jacobian of all the constraints.

The derivation of optimality conditions for NIP requires that the constraint functions satisfy a constraint qualification at x^*. We will assume that $A(x^*)$ has full row rank, which implies that the constraint qualification holds at x^*. Necessary and sufficient conditions for a feasible point x^* to be a local minimizer of NIP are given in the following theorems (see, e.g., Gill *et al.* [GMW81]):

THEOREM 3.1.1. *(First-order necessary optimality conditions.) If $A(x^*)$ has full row rank, a necessary condition for a feasible point x^* to be a minimizer of NIP is*

that there exist a vector λ^* *such that*

$$g(x^*) = A(x^*)^T\lambda^*, \quad \textit{with} \quad \lambda^* \geq 0. \tag{3.1.2}$$

THEOREM 3.1.2. *(Sufficient optimality conditions.) Assuming* $A(x^*)$ *has full row rank and* $Z(x^*)$ *is a basis for the null space of* $A(x^*)$*, a feasible point* x^* *is a strong local minimizer of NIP if there exists a vector* λ^* *such that*

(i) $g(x^*) = A^T(x^*)\lambda^*$;

(ii) $\lambda^* > 0$;

(iii) $Z(x^*)^T\nabla^2 L(x^*, \lambda^*)Z(x^*)$ *is positive definite.*

Condition (ii) of Theorem 3.1.2—that the Lagrange multipliers corresponding to active constraints are strictly positive—is usually termed *strict complementarity.* If any multiplier corresponding to an active constraint is zero, the optimality conditions become more complicated. (For details, see, e.g., Fiacco and McCormick [FM68].)

3.2 Barrier Functions

Barrier-function methods constitute a class of sequential minimization methods for solving the inequality constrained problem NIP (3.1.1), where the feasible region has a nonempty interior. They are characterized by requiring strict satisfaction of all constraints at the initial estimate of the solution and subsequent iterates. This can be advantageous if the objective function is not defined outside the feasible region, or if only a solution of limited accuracy is required.

A barrier-function method creates a sequence of modified functions having minimizers that are strictly feasible to the constraints of NIP. This property is achieved by introducing a weighted *barrier* term, which is a continuous function with a positive singularity at the constraint boundaries. As the weight assigned to the barrier term is decreased towards zero, the minimizers of the barrier function tend to minimizers of the original problem. Since barrier methods generate strictly feasible iterates, they belong to the class of so-called *interior-point methods.*

Many barrier functions have been considered in the context of nonlinear optimization [FM68]. In this dissertation we consider only the *logarithmic* barrier function originally suggested by Frisch, 1955 [Fri55].[1] Application of the logarithmic barrier

[1] We note that Frisch did not minimize the barrier function directly, but instead used its gradient in combination with the objective function to enforce feasibility.

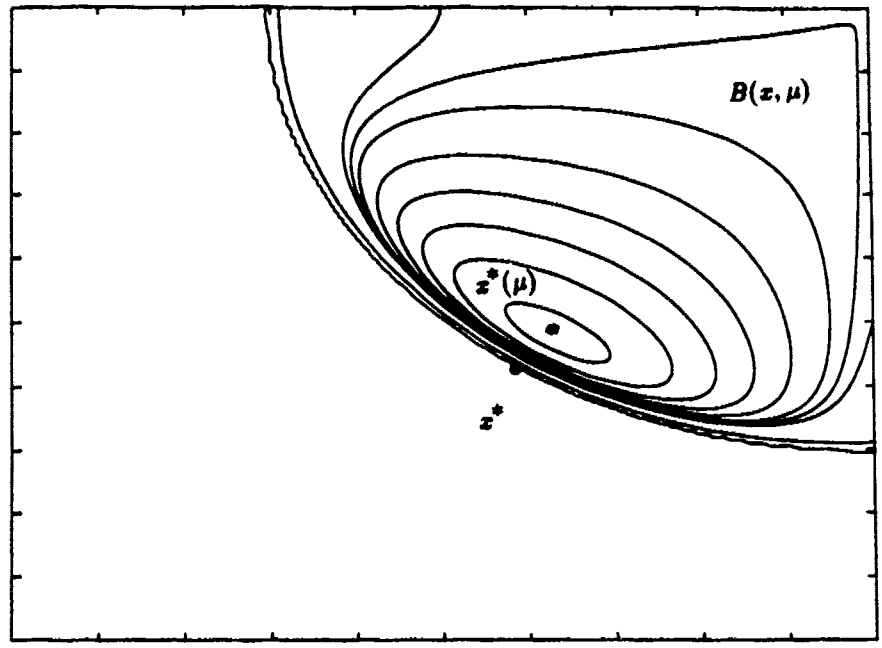

Figure 3.1: Contours of $B(x,\mu)$ for $\mu = 0.1$

transformation to NIP yields the following subproblem:

$$\text{BNP} \qquad \underset{x\in\Re^n}{\text{minimize}} \qquad B(x,\mu) = F(x) - \mu\sum_{i=1}^{m}\ln(c_i(x)) \tag{3.2.1}$$

where the positive scalar μ is known as the *barrier parameter.*

Some fundamental issues have to be addressed when using a barrier method. For example, how well does the unconstrained problem approximate the constrained one?

Typical graphs of the contours of $B(x,\mu)$ for two values of μ are illustrated in Figures 3.1 and 3.2. They exhibit some of the features of barrier functions. As expected, the effects of the logarithmic terms extend further into the interior when μ is large. When μ is small their effect is negligible away from a small neighborhood of the constraint boundaries. By inspection it is clear that for any positive value of μ, the minimum of $B(x,\mu)$ can never lie on a constraint of the original problem. It can also be observed that the iterates $x^*(\mu)$ approach x^* as $\mu \to 0$. This behavior is expressed formally in Theorem 3.3.1.

Notice also that as μ decreases the contours of the corresponding barrier function become more elongated, almost parallel. Hence, we can foresee the difficulty a method might encounter in computing accurately the component of the search direction that is tangential to the constraints active at x^*.

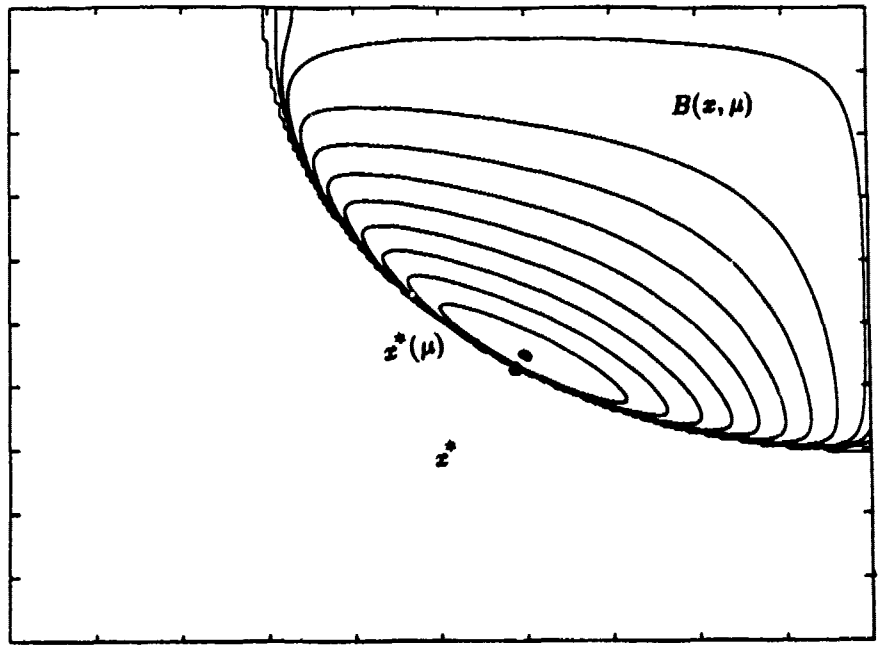

Figure 3.2: Contours of $B(x,\mu)$ for $\mu = 0.01$

3.3 Convergence Results

We shall consider the behavior of the barrier function along a given trajectory. The first theorem shows that all feasible minimizers of $B(x,\mu)$ are related to those of NIP as $\mu \to 0$.

THEOREM 3.3.1. *Let χ^* be the set of minimizers of NIP (3.1.1), and $\chi^*(\mu)$ the set of feasible minimizers for the barrier subproblem BNP (3.2.1). Let $x^*(\mu)$ be any point in $\chi^*(\mu)$ and let x_M^* be the closest point to $x^*(\mu)$ in χ^*. If conditions* (i)-(iii) *of Theorem 3.1.2 hold then*

$$\lim_{\mu \to 0} \|x^*(\mu) - x_M^*\| = 0.$$

Proof. See Fiacco and McCormick [FM68].

Convergence Analysis.

For completeness we present a result equivalent to the one stated in Theorem 3.3.1, but for the special case of strictly convex quadratic programs.

THEOREM 3.3.2. *Consider the following quadratic program:*

$$\begin{aligned} \underset{x \in \Re^n}{\text{minimize}} \quad & f_p(x) = c^T x + \tfrac{1}{2} x^T H x \\ \text{subject to} \quad & Ax = b, \qquad x \geq 0, \end{aligned} \tag{3.3.1}$$

where A is an $m \times n$ matrix of full row rank. We shall assume that

(i) *strict complementarity holds at the solution;*

(ii) *a strictly feasible point exists;*

(iii) *the feasible region is compact;*

(iv) *the matrix H is positive definite.*

Then,

$$\lim_{\mu \to 0} x^*(\mu) = x^*,$$

where x^* is the solution of the QP and $x^*(\mu)$ is the minimizer of

$$\begin{aligned} \underset{x \in \Re^n}{\text{minimize}} \quad & B(x,\mu) = c^T x + \tfrac{1}{2} x^T H x - \mu \sum_{j=1}^{n} \ln x_j \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

Proof. Let $D = \text{diag}(x_j)$ be defined for any $x > 0$, and let Z be a basis for the null space of A. If H is positive definite, the reduced Hessian for the barrier subproblem, namely $Z^T(H + \mu D^{-2})Z$, must also be positive definite for any $\mu > 0$ and $x > 0$. Hence the minimizer $x^*(\mu)$ and the associated Lagrange multipliers $\pi^*(\mu)$ exist and are unique. From the first-order necessary conditions (3.1.2) for the barrier subproblem (3.3.1), we have that

$$\nabla_x L = \nabla_x B(x^*(\mu), \mu) - A^T \pi^*(\mu) = 0,$$

i.e.,

$$c + Hx^*(\mu) - \mu D^{-1} e - A^T \pi^*(\mu) = 0 \tag{3.3.2}$$

where $L \equiv B(x,\mu) - \pi^T(Ax - b)$. Premultiplying this equation by $x^*(\mu)$ gives

$$c^T x^*(\mu) + x^*(\mu)^T H x^*(\mu) - \mu e^T D^{-1} x^*(\mu) - x^*(\mu)^T A^T \pi^*(\mu) = 0.$$

Since $Ax^*(\mu) = b$,

$$c^T x^*(\mu) + x^*(\mu)^T H x^*(\mu) - b^T \pi^*(\mu) = n\mu. \tag{3.3.3}$$

Defining $x^*(0) = \lim_{\mu \to 0} x^*(\mu)$, we see that

$$c^T x^*(0) + x^*(0)^T H x^*(0) - b^T \pi^*(0) = 0. \tag{3.3.4}$$

The dual of the quadratic program considered is

$$\begin{aligned} \underset{\pi,w}{\text{maximize}} \quad & f_d(w,\pi) = b^T\pi - \tfrac{1}{2}w^THw \\ \text{subject to} \quad & A^T\pi - Hw \leq c. \end{aligned} \tag{3.3.5}$$

Since $\mu D^{-1}e > 0$, it follows from (3.3.2) that $(x^*(\mu), \pi^*(\mu))$ is dual feasible. In particular, $(x^*(0), \pi^*(0))$ is dual feasible. Also, the duality gap is

$$f_p(x) - f_d(w,\pi) = c^Tx + \tfrac{1}{2}x^THx - b^T\pi + \tfrac{1}{2}w^THw,$$

so that

$$f_p(x^*(0)) - f_d(x^*(0),\pi^*(0)) = c^Tx^*(0) + x^*(0)^THx^*(0) - b^T\pi^*(0) = 0,$$

using (3.3.4). It follows from (2.5.9) that $x^* = x^*(0)$.

∎

Existence of an Isolated Trajectory.

Under suitable assumptions, the set of feasible minimizers of the barrier function can be regarded as a function of an independent variable μ, tracing out a smooth *trajectory* of points $x^*(\mu)$ converging to x^*.

The assumptions needed to define a unique trajectory of local minima are stronger than those needed to prove the existence of points converging to a local minimizer of NIP. We are interested in an *isolated trajectory*, i.e., a trajectory that is *locally unique*. An isolated trajectory is a continuous function $x^*(\mu)$, where every point $x^*(\bar{\mu})$ on the trajectory is a feasible isolated local minimum of $B(x,\bar{\mu})$. The following result is proved in Fiacco and McCormick [FM68].

THEOREM 3.3.3. *If $A(x^*)$ has full row rank and the sufficient conditions of Theorem 3.1.2 hold at x^*, then for sufficiently small μ there exists a continuously differentiable trajectory $x^*(\mu)$ such that*

$$\lim_{\mu\to 0} x^*(\mu) = x^*,$$

where $x^*(\mu)$ is a local minimizer of $B(x,\mu)$.

Figure 3.3 depicts the trajectory of minimizers of the barrier function applied to the convex QP problem given in Chapter 1.

The trajectory $x^*(\mu)$ has several interesting properties. Expanding about $\mu = 0$ gives the following expression:

$$x(\mu) = x^* + \mu y + O(\mu^2),$$

where

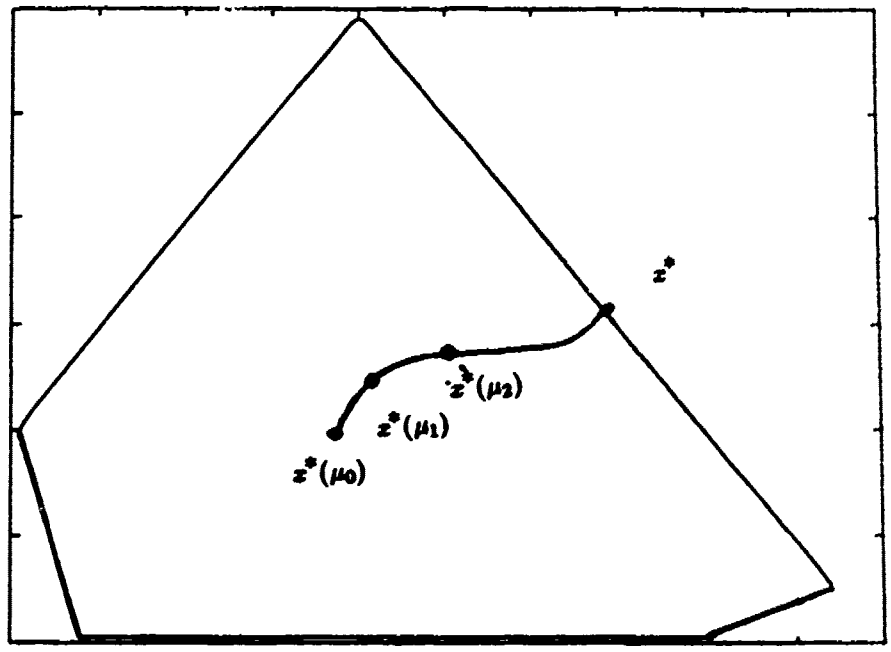

Figure 3.3: Trajectory of $x^*(\mu)$, with $\mu_0 > \mu_1 > \mu_2$

$$y = \lim_{\mu \to 0} \frac{x(\mu) - x^*}{\mu} = \left. \frac{dx(\mu)}{d\mu} \right|_{\mu=0}. \tag{3.3.6}$$

By definition of an unconstrained minimizer, the following relation holds at $x^*(\mu)$:

$$\nabla B = g - \mu \sum_{i=1}^{m} \frac{1}{c_i} a_i = g - A^T \begin{pmatrix} \mu/c_1 \\ \vdots \\ \mu/c_m \end{pmatrix} = 0. \tag{3.3.7}$$

Differentiating (3.3.7) and using (3.3.6), we obtain the following expression:

$$Ay = \begin{pmatrix} 1/\lambda_1^* \\ \vdots \\ 1/\lambda_m^* \end{pmatrix}. \tag{3.3.8}$$

The relationship (3.3.8) implies that the minimizers of successive barrier functions do not approach x^* tangentially to any constraint for which $0 < \lambda_i^* < \infty$.

Existence of Derivatives of $x(\mu)$ for $\mu > 0$.

In general, the trajectory has an order of differentiability with respect to μ (when $\mu > 0$) that is one less than that of the original problem functions and is analytic when the functions are analytic. It has also been shown [FM68] that under certain assumptions the limits of the derivatives at $\mu = 0$ exist and are finite. This has some computational implications.

3.4 Properties of Barrier Functions

Let $\{\mu_k\}$ be a strictly decreasing positive sequence, with $\lim_{k\to\infty} \mu_k = 0$. The minimizers of successive barrier functions along a given trajectory exhibit the following properties:

(i) $\{B_k\}$ is strictly decreasing for sufficiently small μ_k and bounded c_k;

(ii) $\{F_k\}$ is nonincreasing;

(iii) $-\sum_{i=1}^{m} \ln\{c_i(x^*(\mu_k))\}$ is nondecreasing,

where B_k denotes $B(x^*(\mu_k), \mu_k)$, F_k denotes $F(x^*(\mu_k))$ and c_k denotes $c(x^*(\mu_k))$. Property (iii) does *not* imply that all constraint values decrease at successive $x^*(\mu_k)$. A reduction in the barrier parameter allows the constraints to approach the boundary of the feasible region, but does not enforce a decrease in the components of c_k.

Since $\mu > 0$ and $c_i > 0$ for all i, identity (3.3.7) shows that the gradient of F at $x^*(\mu)$ is a *nonnegative linear combination* of *all* the constraint gradients, where the coefficient of a_i is μ/c_i. As μ approaches zero, the quantity $\mu/c_i(x^*(\mu))$ will converge to zero if c_i is not active at x^*, since c_i is strictly bounded away from zero in a neighborhood of x^*. Assume that $\hat{m}$ constraints are active at x^*. Then for sufficiently small μ, the relation holding at $x^*(\mu)$ can be written as

$$g = A^T \begin{pmatrix} \mu/\hat{c}_1 \\ \vdots \\ \mu/\hat{c}_{\hat{m}} \end{pmatrix} + O(\mu), \tag{3.4.1}$$

where $\hat{c}_i$ denotes the i-th active constraint, and A denotes the $\hat{m} \times n$ matrix of active constraint gradients.

It follows from (3.4.1) that the quantity

$$\lambda_i(\mu) \equiv \frac{\mu}{\hat{c}_i(x^*(\mu))}, \tag{3.4.2}$$

defined only for the active constraints, satisfies a relationship with g and A analogous to the multiplier relation that must hold at x^*. The vector $\lambda(\mu)$ satisfies

$$\lambda(\mu) = \lambda^* + O(\mu),$$

where λ^* is the vector of Lagrange multipliers at x^*.

3.5 Barrier Methods

At first sight, it might seem that the solution to NIP could be found simply by setting μ to a very small value and using a standard unconstrained method. Unfortunately, there are difficulties associated with computing an unconstrained minimum of (3.2.1) for a small value of μ. Also, for a given problem it may not be known a priori what the relatively small value of μ should be. Hence a *sequence* of subproblems with decreasing values of the barrier parameter must be solved. The choice of values for μ constitutes an interesting subject of study. The efficiency of the approach critically depends upon a sensible strategy for choosing μ. We are also required to solve an unconstrained problem whose objective function contains a barrier term. Such problems are difficult to solve for several well known reasons.

Most practical barrier methods proceed as a classical continuation algorithm, where the solution from the previous minimization problem is used as an initial estimate of the solution for the next problem. If an accurate solution of a subproblem is found it has been shown (e.g., see [FM68]) that advantage may be taken of the asymptotic behavior of $h = x^*(\mu_k) - x^*$ to estimate $x^*(\mu_{k+1})$, the solution for the next barrier subproblem.

3.6 A Model Algorithm

In this section we shall outline a model algorithm for solving

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & F(x) \\ \text{subject to} \quad & Ax = b \\ & c(x) \geq 0, \end{aligned} \tag{3.6.1}$$

using a barrier function. Assume the following quantities are known: a strictly feasible initial point x_0 ($Ax_0 = b$, $c(x_0) > 0$), a sequence $\{\mu_k\}$ such that $\lim_{k\to\infty} \mu_k = 0$ and $\mu_k > \mu_{k+1} > 0$, and a sequence $\{\delta_k\}$, where $M > \delta_k > 0$.

Algorithm BARALG (Model Barrier Algorithm)

BA1. [Initialize] Set $\bar{x}_0 = x_0$, $k = 0$.

BA2. [Solve the barrier subproblem] Using x_k as the first point, generate a set of strictly feasible points $\{\bar{x}_s(\mu_k)\}$, $s = 0, 1, \ldots, s_k$, that estimate a solution $x^*(\mu_k)$ of the subproblem

$$\begin{aligned} \underset{x \in \mathcal{R}^n}{\text{minimize}} \quad & B(x, \mu_k) = F(x) - \mu_k \sum_{i=1}^{m} \ln c_i(x) \\ \text{subject to} \quad & Ax = b, \end{aligned} \tag{3.6.2}$$

where $\bar{x}_{s_k}(\mu_k)$ is the only point of the set that satisfies the conditions

$$\|Z^T \nabla B(\bar{x}_{s_k}(\mu_k), \mu_k)\| \leq \delta_k \mu_k, \tag{3.6.3}$$

$$\lambda_k \geq -\delta_k \mu_k, \tag{3.6.4}$$

where λ_k is the smallest eigenvalue of $Z^T \nabla^2 B(\bar{x}_{s_k}(\mu_k), \mu_k) Z$, and Z is a matrix whose columns span the null space of A.

BA3. [Update the estimate of the solution] Set $x_{k+1} = \bar{x}_{s_k}(\mu_k)$.

BA4. [Update the iteration count] Set $k = k + 1$ and return to step **BA2**.

Since $\lim_{k \to \infty} \mu_k = 0$, it is self-evident that $\lim_{k \to \infty} \|x_k - x_k^*\| = 0$, where x_k^* is the nearest point to x_k in χ^* (the set of minimizers of 3.6.1). The difficulty lies in being able to generate the set of points $\bar{x}_j(\mu_k)$ so that a point satisfying the relevant conditions will eventually be found. This issue is addressed in the next chapter.

Chapter 4

Solution of Barrier Subproblems

4.1 Introduction

In the previous chapter we described a barrier algorithm. The key step was to determine a point that satisfies conditions (3.6.3) and (3.6.4) for each value of μ. A minimizer of the barrier subproblem is one such point. Moreover, all points in a finite neighborhood of a minimizer satisfy these conditions. Therefore a means of finding the required point is to use an algorithm that finds a minimizer of the barrier subproblem. Since all points in a finite neighborhood satisfy the conditions, any algorithm that generates a subsequence converging to a minimizer will find a suitable point in a finite number of iterations.

A difficulty with general nonlinear minimization is that no known practical algorithms are guaranteed to find minimizers. The best that can be hoped for is an algorithm to find a point that satisfies the necessary conditions for optimality. We can assume that any such point is a minimizer, or accept the fact that the barrier algorithm may converge to a point that satisfies only the necessary conditions. This is no less satisfactory than for current active-set methods. Note that for the original QP, any point satisfying the necessary conditions *is* a minimizer provided all active constraints have nonzero multipliers, and the only difficulty is distinguishing dead-points from minimizers (see Section 2.3).

In this chapter we discuss algorithms for finding minimizers of the barrier subproblem (3.6.2) arising in step **BA2** of the model barrier algorithm BARALG. It will be seen that for the large-scale case, such algorithms are both rare and complex. We also propose an alternative to a general-purpose algorithm that takes specific advantage of the form of the barrier subproblem.

Whatever approach is used, the subproblem has some special difficulties not present in the usual equality-constrained optimization problem. For example, with a normal problem it is trivial to provide an initial point that satisfies the linear equality constraints, but here the point must also be strictly interior with respect to the bounds. If a trust-region approach is used [DS83,Fle87], it would imply that the usual trust-region subproblem has hidden constraints, which may therefore not be satisfied at the solution. Consequently, it may be necessary re-solve the trust-region subproblem. We have restricted our attention to linesearch methods.

4.2 Optimality Conditions

Barrier subproblems in algorithm BARALG have the following form:

$$\begin{array}{lll} \text{LEP} & \underset{x \in \mathbb{R}^n}{\text{minimize}} & F(x) \\ & \text{subject to} & Ax = b, \end{array} \tag{4.2.1}$$

where $F(x)$ is a twice continuously differentiable function and A is an $m \times n$ matrix of full row rank. Let $g(x)$ and $H_F(x)$ denote $\nabla F(x)$ and $\nabla^2 F(x)$ respectively.

We begin by stating the optimality conditions for LEP. The necessary conditions for x^* to be a solution of LEP are given in the following theorem.

THEOREM 4.2.1. *(Necessary conditions for a solution of LEP). If A has full row rank and Z is a basis for the null space of A, necessary conditions for a feasible point x^* to be a minimizer of LEP are*

(i) *there exists a vector π^* such that $g(x^*) = A^T\pi^*$, or equivalently, $Z^T g(x^*) = 0$; and*

(ii) *$Z^T H_F(x^*) Z$ is positive semidefinite.*

The vector $Z^T g(x)$ is termed the *reduced gradient* of F at x. Any point $\bar{x}$ such that $Z^T g(\bar{x}) = 0$ is termed a *constrained stationary point* of F (with respect to the constraints $Ax = b$). The matrix $Z^T H_F(x) Z$ is called the *reduced Hessian.*

Sufficient conditions for optimality are analogous to those for NIP (see Theorem 3.1.2), except that no sign restriction on π^* is imposed and the Hessian of the Lagrangian becomes the Hessian of $F(x)$ itself.

THEOREM 4.2.2. *(Sufficient conditions for an isolated solution of LEP). If A has full row rank and Z is a basis for the null space of A, a feasible point x^* is an isolated solution of LEP if*

(i) *there exists a vector π^* such that $g(x^*) = A^T\pi^*$, or equivalently, $Z^T g(x^*) = 0$; and*

(ii) *$Z^T H_F(x^*) Z$ is positive definite.*

If x^* satisfies the necessary conditions of Theorem 4.2.1, the problem of determining whether or not it is an isolated minimizer is an NP-hard problem.

4.3 Methods for Solving LEP

Almost all current methods for solving LEP are feasible descent methods. Given an initial feasible point, a sequence $\{x_k\}$ is generated that retains feasibility and for which $\{F(x_k)\}$ is strictly monotonically decreasing. Normally all function values $F(x_k)$ must be computed. For a logarithmic barrier function it means evaluating many logs. While this is not prohibitively expensive it is not wholly desirable. We shall discuss means of avoiding such computations.

We shall be interested in *linesearch methods*, in which a step s is chosen to achieve a "sufficient" decrease in some merit function. Initially we assume that the current point x satisfies $Ax = b$, and we use $F(x)$ itself as a merit function. The next iterate $x + s$ must satisfy $A(x + s) = b$, and $F(x + s)$ must be sufficiently less than $F(x)$. The exact meaning of "sufficiently less" will be made clear later.

Ordinarily, obtaining a feasible point to a set of linear equality constraints is a trivial problem. However, in our case we require an initial point that satisfies $Ax = b$ and $x > 0$. We shall therefore consider an approach that does not require the initial point to be feasible with respect to the linear equality constraints.

We shall restrict our interest to Newton's method and its many variants.

4.4 Transformation to an Unconstrained Problem

We may solve LEP by first transforming it to an unconstrained problem. Suppose a point x_0 is known such that $Ax_0 = b$. A solution of LEP is given by $x_0 + Zy^*$, where y^* is a minimizer of the function $\mathcal{F}(y) \equiv F(x_0 + Zy)$. If Newton's method is applied to the problem

$$\min_y \mathcal{F}(y),$$

the search direction in the y-space, say p_z, is given as the solution of the following equations:

$$Z^T H_F Z p_z = -Z^T g. \tag{4.4.1}$$

The equivalent search direction in the x-space is given by $p = Zp_z$. Clearly this approach is practical only if it is computationally convenient to form the products $Z^T H_F Z$ and $Z^T g$ for some appropriate matrix Z.

Many methods have been proposed to solve unconstrained minimization problems (see, e.g., Fiacco and McCormick [FM68], Gill and Murray [GM74], McCormick [McC77], Fletcher and Freeman [FF77], Mukai and Polak [MP78], Kaniel and Dax [KD79], Moré and Sorensen [MS79], Goldfarb [Gol80] and Forsgren, Gill and Murray [FGM89a]) and some can be extended to the large-scale case. The methods are based mainly on computing a direction of sufficient descent and a direction of sufficient negative curvature whenever such directions exist. The methods vary on how to compute such directions. Typically a descent direction is determined by first identifying a

matrix E such that $Z^T H_F Z + E$ is positive definite. The modified matrix is then used in place of $Z^T H_F Z$ in (4.4.1). Computing a direction of negative curvature is often more difficult but a number of methods are known [McC77,MS79,Gol80,FGM88].

In some applications such as structural analysis, A is such that a *sparse* null-space matrix Z can be found. If H_F is diagonal and positive definite (as in the barrier subproblems for the LP case), it is conceivable that $Z^T H_F Z$ might also be sparse, and that current sparse Cholesky factorization [GL81,CGLN84] methods could be applied.

For a general H_F, however, $Z^T H_F Z$ is not likely to be sparse, and a direct factorization will not be practical unless $n - m$ (the column dimension of Z) is quite small (say ≤ 50). The difficulty is not the need to store a dense matrix but rather the considerable effort require to form the matrix product. Some economy is possible in the case of the barrier function when all the inequalities are simple bounds. The Hessian is then of the form $H_z = H + D$, where D is a diagonal matrix. Only D is a function of x, and we may form $Z^T H_F Z$ by forming $Z^T H Z$ only once. If Z is sufficiently sparse (as in the structural analysis example), each subsequent matrix $Z^T D Z$ may be sparse and cheap to form.

If H_F and Z have the property that products of the form $H_F u$, Zv and $Z^T w$ can be computed efficiently, a conjugate-gradient method [HS52,GL89] could be applied. If $Z^T H_F Z$ is positive-definite for all relevant values of $H_F(x)$, the theory of inexact Newton methods [DES82] shows that p_z need not be computed accurately (but the accuracy should increase as $x \to x^*$).

In general, conjugate-gradient methods are likely to require too many iterations unless the system involved has a low condition number or a favorable clustering of eigenvalues. Sometimes a preconditioner $M = CC^T$ might be known that induces such properties in the transformed system $C^{-1}(Z^T H_F Z)C^{-T}$. (See [GL89] for a discussion of preconditioners.) However, there is no general procedure for finding a good preconditioner in this context.

4.5 The KKT System

It has been shown that while a null-space approach may be possible for some problems it is not viable for general problems. Rather than using (4.4.1), a mathematically equivalent way to obtain p is to solve the KKT system

$$K \begin{pmatrix} -p \\ \pi \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \tag{4.5.1}$$

where

$$K = \begin{pmatrix} H_F & A^T \\ A & 0 \end{pmatrix}.$$

Recently Forsgren and Murray [FM90] have proposed a method based on a direct factorization of K. The approach we have taken in our implementation is similar and involves the same matrix K (see Section 4.7). When A and H_F are sparse, it is reasonable to suppose that a sparse factorization of K exists.

When $Z^T H_F Z$ is not available, the many methods to obtain both a descent direction and a direction of negative curvature are no longer applicable. When $Z^T H_F Z$ is not positive definite suitable descent direction can be computed by replacing this matrix by *any* posi' definite matrix. If the inertia of K is known, we can determine when $Z^T H_F Z$ is n positive definite by applying the result of Lemma 2.1.2. Unfortunately if K does not have the correct inertia, to obtain a suitable direction it is not sufficient to replace K with just *any* matrix with suitable inertia. More significantly, it was not clear until the work of Forsgren and Murray how to compute a direction of negative curvature efficiently knowing only a factorization of K.

4.6 The LBL^T Factorization

Since K is symmetric but indefinite, the preferred method of determining a solution to (4.5.1) is via the factorization $P^T K P = LBL^T$, where P is a permutation matrix, L is unit lower triangular, and B is block diagonal with blocks of dimension 1 or 2; see Bunch and Parlett [BP71], Bunch and Kaufman [BK77].

The LBL^T factorization in the large-scale case is performed in two stages: an analyze phase and a numerical phase. The analyze phase is a symbolic factorization that determines a pivoting strategy (a row and column ordering) to minimize the fill-in in the factor L. Since the sparsity pattern of the KKT systems is constant, the analyze phase is needed only once. A strategy commonly used is the minimum-degree ordering, which seeks to reduce fill-in by choosing a diagonal pivot corresponding to the row that currently has the least number of nonzero elements. In the numerical phase the pivot order may be changed to ensure numerical stability. A consequence of the second level of ordering is that $\|L\|$ is bounded. Forsgren and Murray [FM90] have suggested a third level of ordering, in which the objective is to ensure that the inertia of the partially formed factorization satisfies certain rules. The pivoting strategy they propose is shown to be both necessary and sufficient for computing a direction of sufficient descent and a direction of sufficient negative curvature (when such a direction exists) using the LBL^T factorization.

Having obtained a direction of sufficient descent and a direction of sufficient negative curvature, Forsgren and Murray suggest that the two directions be used in a curvilinear linesearch, or combined in a particular fashion for a regular linesearch. In our case we have a strong preference for a regular linesearch, since it is imperative that the steplength be chosen to ensure that the new iterate is feasible with respect to the constraints of the original problem. The existence of such a point is guaranteed,

since $F(x)$ and its derivatives are infinite at the boundary of the original feasible region. No matter which of the standard steplength criteria are used, a strictly feasible point satisfying them exists.

Although the method of Forsgren and Murray could be applied to the subproblem, it would be useful if we could avoid introducing the third level of ordering. Moreover, it would be beneficial if some merit function other than the barrier function could be used. The method of Forsgren and Murray also requires an initial feasible point (for the original constraints) and in general such a point will not be known. In the next sections we describe an approach that attempts to circumvent these difficulties.

4.7 Newton's Method Applied to the Optimality Conditions

The first-order optimality conditions for LEP can be expressed by the nonlinear equations

$$\begin{aligned} g(x) - A^T\pi &= 0, \\ Ax - b \quad &= 0. \end{aligned} \tag{4.7.1}$$

In order to simplify the problem of finding an initial point satisfying $Ax = b$, we have chosen to treat these equations directly. Newton's method applied to these nonlinear equations leads to a KKT system similar to (4.5.1).

Let the optimality equations (4.7.1) be written as

$$f(x,\pi) \equiv \begin{pmatrix} g(x) - A^T\pi \\ Ax - b \end{pmatrix} = 0. \tag{4.7.2}$$

Newton's method applied to (4.7.2) corresponds to the iteration

$$\begin{aligned} J(x_k,\pi_k)\begin{pmatrix} p \\ q \end{pmatrix} &= -f(x_k,\pi_k), \\ \begin{pmatrix} x_{k+1} \\ \pi_{k+1} \end{pmatrix} &= \begin{pmatrix} x_k \\ \pi_k \end{pmatrix} + \alpha \begin{pmatrix} p \\ q \end{pmatrix}, \end{aligned} \tag{4.7.3}$$

where $J(x_k,\pi_k)$ is the Jacobian of $f(x,\pi)$ at (x_k,π_k) and $0 < \alpha \leq 1$. The linear system in (4.7.3) is analogous to the KKT system for optimization with nonlinear objective functions, and is of the form

$$\begin{pmatrix} H_F(x_k) & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} -p \\ q \end{pmatrix} = \begin{pmatrix} g_L \\ r \end{pmatrix}, \tag{4.7.4}$$

where

$$g_L = g(x_k) - A^T\pi_k$$

is the gradient of the Lagrangian for LEP, and

$$r = Ax_k - b$$

is the constraint residual. If the Jacobian at x^* is nonsingular, Newton's method converges quadratically to a stationary point when (x_0, π_0) is sufficiently close to the solution.

In the following we shall use the notation (p, q) to denote the column vector $(p^T\ q^T)^T$.

LEMMA 4.7.1. *When the vector (p, q) is obtained via (4.7.3) and a sufficiently small α is chosen for the steplength, every nonzero component of the vector (g_L, r) decreases.*

Proof. The proof of this fundamental property of Newton's method follows from the Taylor series expansion of $f(x_k, \pi_k)$.

Define the following notation: $f = f(x_k, \pi_k)$, $\bar{f} = f(x_{k+1}, \pi_{k+1})$, $r = Ax_k - b$, $\bar{r} = Ax_{k+1} - b$, $g_L = \nabla F(x_k) - A^T\pi_k$, $\bar{g}_L = \nabla F(x_{k+1}) - A^T\pi_{k+1}$, and $J = J(x_k, \pi_k)$. Then

$$\bar{f} = f + \alpha J \begin{pmatrix} p \\ q \end{pmatrix} + t(\alpha), \tag{4.7.5}$$

where the $t(\alpha)$ term is $O(\alpha^2)$. From equations (4.7.3) and (4.7.5) we have

$$\bar{f} = (1 - \alpha)f + t(\alpha).$$

For (4.7.1) this implies

$$\begin{aligned} \bar{g}_L &= (1 - \alpha)g_L + t(\alpha), \\ \bar{r} &= (1 - \alpha)r. \end{aligned} \tag{4.7.6}$$

For a sufficiently small α, $t(\alpha)$ is negligible and the result follows.

∎

The lemma implies that every nonzero component of r decreases for any value of $\alpha \in (0, 1]$. Zero components of r will remain zero, but those of g_L may become nonzero, according to $t(\alpha)$. We note that when Newton's method is applied to the specific barrier functions considered in this thesis, an explicit expression for $t(\alpha)$ can be determined (see Section 4.9).

Although Lemma 4.7.1 seems a satisfying result, a very *small* step may be needed if we required all components of f to decrease. A more useful result is given in the following lemma.

LEMMA 4.7.2. *When the vector (p,q) is obtained via (4.7.3) and a sufficiently small α is chosen for the steplength, $\|(g_L, r)\|_2$ decreases.*

Proof. The required result is a specific case of a standard result for Newton's method applied to $f(x) = 0$; namely, that the Newton direction generated from $Jp = -f$ is a descent direction for the function $M(x) = \frac{1}{2}\|f\|_2^2$:

$$(\nabla M)^T p = (J^T f)^T p = f^T J p = -f^T f < 0.$$

∎

For the rest of this chapter, $\|.\|$ will denote the Euclidean norm. It follows that a steplength α can be chosen to ensure a decrease in $\|(g_L, r)\|^2$. In other words,

$$M_2(x,\pi) = \|g_L\|^2 + \|r\|^2$$

could serve as a *merit function* for determining α. The choice of α will depend on the linesearch criteria. Almost any criteria will do. For example, for the Goldstein-Armijo conditions [Arm66,Gol67] it can be shown that the sequence $\{x_k\}$ will converge to a stationary point provided the Jacobian matrix is nonsingular at all the iterates. While this property cannot be shown to be true in general, we shall show later that for barrier functions with a judicious choice of $\{\mu_k\}$ and $\{\delta_k\}$, we can arrange for K to be almost always nonsingular.

A key property of M_2 is that (like F) both the function and its derivatives are infinite on the boundary of the original feasible region. Hence we may choose a steplength satisfying standard criteria that ensures that the new iterate remains strictly feasible with respect to the original constraints.

4.8 An Alternative Merit Function

For our specific nonlinear equations (4.7.1), we shall consider an alternative merit function,[1] namely

$$M(x,\pi) = \|g_L\| + \|r\|. \tag{4.8.1}$$

LEMMA 4.8.1. *Let $M(x,\pi), p$ and q be defined by (4.8.1) and (4.7.4) respectively. Then $\nabla M(x,\pi)$ projected along (p,q) satisfies*

$$(p^T \quad q^T)\nabla M(x,\pi) = -\|g_L\| - \|r\|.$$

[1] Note that a decrease in the merit function does not necessarily correspond to a decrease in the objective function $F(x)$. However, if the reduced Hessian of $F(x)$ is positive definite and $r = 0$, the search direction is also a direction of descent for the objective function.

Proof. Rewriting $M(x,\pi)$ as

$$M(x,\pi) = (g_L^T g_L)^{1/2} + (r^T r)^{1/2}, \tag{4.8.2}$$

we have

$$\begin{aligned} \frac{\partial M(x,\pi)}{\partial x} &= \tfrac{1}{2}(g_L^T g_L)^{-1/2}\frac{\partial g_L^T g_L}{\partial x} + \tfrac{1}{2}(r^T r)^{-1/2}\frac{\partial r^T r}{\partial x}, \\ \frac{\partial M(x,\pi)}{\partial \pi} &= \tfrac{1}{2}(g_L^T g_L)^{-1/2}\frac{\partial g_L^T g_L}{\partial \pi}. \end{aligned} \tag{4.8.3}$$

We can simplify the terms in (4.8.3) by differentiating:

$$\begin{aligned} \frac{\partial g_L^T g_L}{\partial x} &= 2(\nabla g_L)^T g_L \\ &= 2(\nabla(g(x) - A^T\pi))^T g_L \\ &= 2H_F(x) g_L \end{aligned} \tag{4.8.4}$$

and

$$\frac{\partial r^T r}{\partial x} = 2A^T r. \tag{4.8.5}$$

From (4.8.3), (4.8.4) and (4.8.5), we now get

$$\begin{aligned} \frac{\partial M(x,\pi)}{\partial x} &= \frac{1}{2 \parallel g_L \parallel}\frac{\partial g_L^T g_L}{\partial x} + \frac{1}{2\parallel r \parallel}\frac{\partial r^T r}{\partial x}, \\ &= H_F(x)\frac{g_L}{\parallel g_L \parallel} + A^T\frac{r}{\parallel r \parallel}. \end{aligned}$$

Repeating a similar procedure for the differentiation with respect to π yields

$$\begin{aligned} \frac{\partial M(x,\pi)}{\partial \pi} &= \frac{1}{2 \parallel g_L \parallel}\frac{\partial g_L^T g_L}{\partial \pi} \\ &= -A\frac{g_L}{\parallel g_L \parallel}. \end{aligned}$$

If we use $\hat{g}_L$ and $\hat{r}$ to denote the normalized vectors g_L and r respectively, the gradient of $M(x,\pi)$ can be written as

$$\nabla M(x,\pi) \equiv \begin{pmatrix} H_F & A^T \\ -A & 0 \end{pmatrix}\begin{pmatrix} \hat{g}_L \\ \hat{r} \end{pmatrix}.$$

Finally, the projected gradient of $M(x,\pi)$ can be determined:

$$\begin{aligned}
(p^T \;\; q^T)\nabla M(x,\pi) &= (p^T \;\; q^T)\begin{pmatrix} H_F & A^T \\ -A & 0 \end{pmatrix}\begin{pmatrix} \hat{g}_L \\ \hat{r} \end{pmatrix} \\
&= (\hat{g}_L^T \;\; \hat{r}^T)\begin{pmatrix} H_F & -A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix} \\
&= \;\; (\hat{g}_L^T \;\; \hat{r}^T)\begin{pmatrix} H_F & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} p \\ -q \end{pmatrix} \\
&= -(\hat{g}_L^T \;\; \hat{r}^T)\begin{pmatrix} H_F & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} -p \\ q \end{pmatrix} \\
&= -(\hat{g}_L^T \;\; \hat{r}^T)\begin{pmatrix} g_L \\ r \end{pmatrix}.
\end{aligned}$$

Hence,

$$(p^T \;\; q^T)\nabla M(x,\pi) = -\|g_L\| - \|r\|.$$

∎

Notice that once x is feasible (i.e. $r = 0$) the merit function becomes

$$M(x,\pi) = \|g_L\|, \tag{4.8.6}$$

and satisfies

$$(p^T \;\; q^T)\nabla\|g_L\| = -\|g_L\|. \tag{4.8.7}$$

4.9 Newton's Method Applied to the Barrier Subproblem

In this section we apply Newton's method to the logarithmic barrier-function subproblems arising from a QP in the following standard form:

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & \frac{1}{2}x^THx + c^Tx \\ \text{subject to} & Ax = b, \quad \ell \le x \le u, \end{array} \tag{4.9.1}$$

where ℓ and u denote upper and lower bounds. The barrier reformulation yields the subproblems

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & B(x,\mu) \equiv c^Tx + \frac{1}{2}x^THx - \mu\sum_{j=1}^{n}\ln(x_j - \ell_j) - \mu\sum_{j=1}^{n}\ln(u_j - x_j) \\ \text{subject to} & Ax = b. \end{array} \tag{4.9.2}$$

Since the equality constraints cannot be treated by a barrier transformation, they are handled directly. Denote

$$\begin{aligned} y_j &= x_j - \ell_j, & z_j &= u_j - x_j, \\ D_\ell &= \operatorname{diag}(1/y_j), & D_u &= \operatorname{diag}(1/z_j), \\ D_g &= D_\ell - D_u, & D_B &= D_\ell^2 + D_u^2. \end{aligned} \tag{4.9.3}$$

To deduce the KKT system corresponding to (4.9.2) we differentiate $B(x,\mu)$ with respect to x_j:

$$\begin{aligned} &\text{(a)} \quad g_B(x) \equiv B_x(x,\mu) = c + Hx - \mu D_g e, \\ &\text{(b)} \quad H_B(x) \equiv B_{xx}(x,\mu) = H + \mu D_B. \end{aligned} \tag{4.9.4}$$

Thus, the KKT matrix K has the form

$$K = \begin{pmatrix} H_B & A^T \\ A & 0 \end{pmatrix} = \begin{pmatrix} H + \mu D_B & A^T \\ A & 0 \end{pmatrix}. \tag{4.9.5}$$

Notice that changes in x and μ only affect the diagonal of the upper block of K. This implies that only the diagonal changes between Newton iterations and between changes to the parameter μ. This fact has significant consequences for both the sensitivity of the linear system (to be discussed in Chapter 5) and the efficient computation of subproblem solutions (see Chapter 8).

We recall from Lemma 4.7.1 that the gradient of the Lagrangian is given by

$$\bar{g}_L = (1-\alpha)g_L + t(\alpha),$$

where $\bar{g}_L$ is the gradient at x_{k+1} and $t(\alpha) = O(\alpha^2)$. In the case of the logarithmic barrier function, we can derive an explicit expression for $t(\alpha)$. Writing $\bar{g}_B \equiv g_B(x_{k+1})$ as a function of $g_B \equiv g_B(x_k)$ yields

$$\begin{aligned} \bar{g}_B &= c + H\bar{x} - \mu \bar{D}_g e \\ &= g + \alpha Hp - \mu \bar{D}_g e \\ &= (g - \mu D_g e) + \mu D_g e + \alpha Hp - \mu \bar{D}_g e \\ &= g_B + \alpha Hp + \mu(D_g - \bar{D}_g)e. \end{aligned} \tag{4.9.6}$$

From (4.7.4) we obtain

$$A^T q = g_L + H_B p, \tag{4.9.7}$$

and from (4.9.7) and the definition of g_L,

$$\begin{aligned} \bar{g}_L &= \bar{g}_B - A^T\bar{\pi} \\ &= \bar{g}_B - A^T\pi - \alpha A^T q \\ &= \bar{g}_B - A^T\pi - \alpha(g_L + H_B p). \end{aligned} \tag{4.9.8}$$

Using expression (4.9.6) for $\bar{g}_B$ and (4.9.8) we have

$$\begin{aligned}\bar{g}_L &= (g_B - A^T\pi) - \alpha g_L + \alpha(Hp - H_B p) + \mu(D_g - \bar{D}_g)e \\ &= (1-\alpha)g_L - \mu\alpha D_B p + \mu(D_g - \bar{D}_g)e.\end{aligned} \tag{4.9.9}$$

The nonlinear term in (4.9.9) is therefore

$$t(\alpha) = \mu(D_g - \bar{D}_g)e - \mu\alpha D_B p.$$

Each component of $t(\alpha)$ can therefore be expressed as

$$\begin{aligned}t(\alpha)_j &= \mu\left(\frac{1}{y_j} - \frac{1}{(y_j+\alpha p_j)} - \frac{1}{z_j} + \frac{1}{(z_j-\alpha p_j)}\right) - \mu\alpha p_j\left(\frac{1}{y_j^2} + \frac{1}{z_j^2}\right) \\ &= \mu\alpha p_j\left(\frac{1}{y_j(y_j+\alpha p_j)} + \frac{1}{z_j(z_j-\alpha p_j)}\right) - \mu\alpha p_j\left(\frac{1}{y_j^2} + \frac{1}{z_j^2}\right) \\ &= \mu\alpha p_j\left(\frac{\alpha p_j}{z_j^2(z_j-\alpha p_j)} - \frac{\alpha p_j}{y_j^2(y_j+\alpha p_j)}\right) \\ &= \mu\alpha^2 p_j^2\left(\frac{1}{z_j^2(z_j-\alpha p_j)} - \frac{1}{y_j^2(y_j+\alpha p_j)}\right).\end{aligned} \tag{4.9.10}$$

For components corresponding to free variables, the nonlinear term is not present and hence g_{L_j} decreases monotonically for any value α in $(0,1]$. For variables with lower (upper) bound only, we see that $t(\alpha)_j$ is large when either the quantity y_j (z_j) or the quantity $y_j + \alpha p_j$ ($z_j - \alpha p_j$) is small. That is, when we are near a bound and μ is not small, the α necessary to guarantee a decrease may be quite small. We shall return to this issue in Chapter 8.

4.10 Backtracking Linesearch

Given the search direction (p, q), we choose a steplength α to reduce the univariate function

$$\phi(\alpha) = M(x + \alpha p, \pi + \alpha q),$$

using a simple *backtracking* linesearch (Dennis and Schnabel [DS83]). Some initial steplength $\alpha = \alpha_0$ is accepted if the following test is satisfied:

$$\phi(\alpha) \leq \phi(0) + \gamma\alpha\phi'(0),$$

where γ is a constant controlling the accuracy of the linesearch. A typical value is $\gamma = 0.5$. Otherwise the test is repeated for the sequence $\{(\frac{1}{2})^i \alpha_0,\ i = 1, 2, \ldots\}$ until the test is satisfied. We have shown in Section 4.8 that the chosen function $M(x, \pi)$ satisfies

$$\phi'(0) = -\phi(0),$$

so that

$$\phi(\alpha) \leq (1 - \gamma\alpha)\phi(0).$$

This result is interesting because it provides a bound on the reduction in $\|f(x, \pi)\|$ at the new point.

The simple backtracking strategy is typically both efficient and convenient for barrier functions. We may choose $\alpha_0 = \min(1,\ \beta\alpha_{\max})$, where $\alpha_{\max}$ is the maximum feasible step along p and $0 < \beta < 1$, and we are then assured of determining a steplength that guarantees the new iterate is strictly feasible. For iterates not close to a minimizer it is often the case that $\beta\alpha_{\max} < 1$. The initial step then usually satisfies the termination test.

4.11 The Inertia of K

It has been essential to the discussion of Newton's method applied to the optimality conditions that K has inertia $(n,\ m,\ 0)$. In general this cannot be guaranteed. In this section we discuss how to ensure that K has the correct inertia in most cases, and what steps may be taken on those occasions when it does not. It should be noted that difficulties can arise only when $Z^T H Z$ is not positive definite.

First note that K can be singular if and only if $Z^T H_B Z$ is singular. Moreover, if $Z^T H_B Z$ is positive definite then K has the required inertia. We can certainly ensure this at the initial point simply by choosing μ_0 sufficiently large, since

$$Z^T H_B Z = Z^T H Z + \mu Z^T D Z$$

where D is positive definite. We shall endeavor to take advantage of this relationship to ensure that K has the required inertia. If μ is large enough, K has the required inertia at all points in the level set $\{\ \Omega \mid M(x, \mu) < M(x_0, \mu)\}$. It follows that convergence to a point satisfying the required conditions is possible for $\mu = \mu_0$. It is nontrivial to select a value of μ_0 that satisfies our criterion. However, there is a simple remedy in practice. If the initial choice for μ_0 proves to be too small (because the inertia of K is incorrect) then μ_0 can be increased until K has the required inertia.

Another useful relationship is that the reduced Hessian is positive definite in the neighborhood of a minimizer at which the reduced Hessian is positive definite. Suppose we are within such a neighborhood. If μ is reduced only slightly, the reduced Hessian of the new barrier function will also be positive definite. This suggests the following strategy for ensuring that K has the correct inertia:

- Select a large value for μ_0 and increase it if necessary until K has the required inertia.

- Whenever μ is reduced, if K does not have the required inertia at the new value of μ, change μ to be less than the old value but greater than its current value.

It also suggests that the criterion for reducing μ should be quite stringent. However, if a large μ_0 is chosen and the neighborhood for which $Z^T H_B Z$ is positive definite is large, it may be possible to reduce μ quite significantly, and without the need to find a close approximation to a minimizer. (See Figures 4.1–4.3.)

Two questions remain to be answered. Firstly, we need to show that a significant reduction in μ is always possible. Secondly, we need to know how to proceed if K does not have the required inertia. Note that the latter is only of concern when $\mu \neq \mu_0$ or it is not the first iteration after μ has been changed.

There is no loss of generality if we assume $r = 0$ at any iteration for which K does not have the required inertia, and if K remains sufficiently nonsingular (so that α is bounded away from zero). Since $r \leftarrow (1-\alpha)r$, $r \to 0$ and once it reaches zero it remains zero.

By having a sufficiently stringent termination criterion for $\mu = \mu_0$, it is always possible to ensure that $r = 0$ for $\mu = \mu_0$ and hence for all $\mu < \mu_0$. Note that even if $r \neq 0$, a point satisfying $r = 0$ would eventually be found provided the weaker condition of K being nonsingular was always satisfied.

Reusing Factors of K.

The discovery that K does not have the correct inertia may be made during the process of computing the LBL^T factorization of K. We could then employ the pivoting strategy suggested by Forsgren and Murray [FM90], but the following lemma suggests that a suitable descent direction or a direction of negative curvature may be obtained by decreasing μ to $\bar{\mu}$ (say) and using the current factors.

LEMMA 4.11.1. *Let p be defined as follows:*

$$\begin{pmatrix} H_B(\mu) & A^T \\ A & \end{pmatrix} \begin{pmatrix} p \\ -q \end{pmatrix} = - \begin{pmatrix} g_L(\bar{\mu}) \\ 0 \end{pmatrix},$$

and assume $p \neq 0$ and $K(\mu)$ is nonsingular. If $\bar{\mu} < \mu$ then at least one of the following inequalities holds:

$$p^T H_B(\bar{\mu}) p < -\gamma \qquad \text{or} \qquad g_B(\bar{\mu})^T p \leq -\gamma,$$

where $\gamma > 0$.

Proof. We have

$$H_B(\mu) = H_B(\bar{\mu}) + (\mu - \bar{\mu})D,$$

where D is a positive definite diagonal matrix. It follows that

$$H_B(\bar{\mu})p + (\mu - \bar{\mu})Dp - A^T q = -g_L(\bar{\mu}).$$

Premultiplying by p^T gives

$$p^T H_B(\bar{\mu})p = -p^T g_B(\bar{\mu}) - (\mu - \bar{\mu})p^T Dp,$$

so that

$$p^T g_B(\bar{\mu}) + p^T H_B(\bar{\mu})p = -\beta,$$

where $\beta = (\mu - \bar{\mu})p^T Dp > 0$.

Clearly, either $p^T g_B(\bar{\mu}) \leq -\frac{1}{2}\beta$ or $p^T H_B(\bar{\mu})p < -\frac{1}{2}\beta$. Setting $\gamma = \frac{1}{2}\beta$ gives the required result.

∎

Note that if $p^T H_B(\bar{\mu})p < -\gamma$ and $p^T g_B(\bar{\mu}) > 0$ then we must have $\bar{p}^T H_B(\bar{\mu})\bar{p} < -\gamma$ and $\bar{p}^T g_B(\bar{\mu}) < 0$, where $\bar{p} = -p$. Thus, if we have a direction of negative curvature we can always construct a direction that is a non-ascent direction *and* a direction of negative curvature.

The significance of this result is two-fold. Firstly, we are able to compute a suitable descent direction and/or a direction of negative curvature. Secondly, we can do so without the need to refactorize the KKT matrix. The assumption that $p \neq 0$ and $K(\mu)$ is nonsingular is not unduly restrictive. If either condition does not hold at x, there exists $\theta \in (0,1)$ such that $K(\theta\mu)$ is nonsingular and $g_L(\theta\mu) \neq 0$. This last result implies $p(\theta\mu) \neq 0$. It is of course necessary to refactorize the KKT matrix if $K(\mu)$ proves to be singular. The following lemma shows that the set of values θ, for which $K(\theta\mu)$ is singular, is finite.

LEMMA 4.11.2. *If $K(x,\mu)$ is singular there exist at most $n-m$ values of θ such that $K(x,\theta\mu)$ is singular.*

Proof. K is singular if and only if $Z^T H_B Z$ is singular (see Forsgren and Murray [FM90]). Now $Z^T H_B(\theta\mu)Z = Z^T H_B(\mu)Z + (\theta - 1)\mu Z^T DZ$, where D is nonsingular and positive definite. Hence

$$\begin{aligned} Z^T H_B(\theta,\mu)Z &= Z^T H_B(\mu)Z + (\theta - 1)R^T R \\ &= R^T(R^{-T} Z^T H_B(\mu) Z R^{-1} + (\theta - 1)I)R. \end{aligned}$$

where

$$R^T R = \mu Z^T DZ.$$

Thus, $Z^T H_B(\theta, \mu) Z$ is singular if and only if $(\theta - 1)$ is equal one of the $n-m$ eigenvalues of $R^{-T} Z^T H_B(\mu) Z R^{-1}$.

∎

Asymptotic Reduction of μ.

We use the notation K/H to denote the *Schur complement of* H for a matrix K partitioned as follows:

$$K = \begin{pmatrix} H & A^T \\ A & M \end{pmatrix}, \qquad K/H \equiv M - AH^{-1}A^T, \tag{4.11.1}$$

where H is assumed to be nonsingular.

To show that we are not inhibited in the asymptotic rate at which μ can be reduced we first need the following lemma.

LEMMA 4.11.3. *If*

$$K_{FR} \equiv \begin{pmatrix} H_{FR} & A_{FR}^T \\ A_{FR} & 0 \end{pmatrix}$$

has inertia $(n_{FR}, m, 0)$, *where* H_{FR} *is an* $n_{FR} \times n_{FR}$ *symmetric matrix and* A_{FR} *is an* $m \times n_{FR}$ *matrix of rank* m, *then there exists* $\gamma < \infty$ *such that the inertia of*

$$K \equiv \begin{pmatrix} H + \gamma I_{FX} & A^T \\ A & 0 \end{pmatrix}$$

is $(n, m, 0)$, *where* H *is an* $n \times n$ *matrix and* A *is an* $m \times n$ *matrix such that*

$$H = \begin{pmatrix} H_{FX} & \tilde{H}^T \\ \tilde{H} & H_{FR} \end{pmatrix},$$

$$A = (A_{FX}, A_{FR}),$$

$$I_{FX} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix},$$

and $\|H\| + \|A\| < \infty$.

Proof. We have

$$In(K) = In(H_{FX} + \gamma I) + In(K/(H_{FX} + \gamma I)). \tag{4.11.2}$$

If $\gamma > -\lambda_{\min}$, where $\lambda_{\min}$ is the smallest eigenvalue of H_{FX}, then

$$In(H_{FX} + \gamma I) = (n_{FX}, 0, 0). \tag{4.11.3}$$

By definition,

$$K/(H_{FX}+\gamma I) = K_{FR} - \begin{pmatrix} \bar{H} \\ A_{FX} \end{pmatrix} (H_{FX}+\gamma I)^{-1}(\bar{H}^T,\ A_{FX}^T),$$

and have

$$\lim_{\gamma\to\infty} \| \begin{pmatrix} \bar{H} \\ A_{FX} \end{pmatrix} (H_{FX}+\gamma I)^{-1}(\bar{H}^T,\ A_{FX}^T)\| = 0.$$

Hence there exists $M < \infty$ such that

$$In(K/(H_{FX}+\gamma I)) = In(K_{FR}) \tag{4.11.4}$$

for $\gamma > M$. It follows from this result, (4.11.2) and (4.11.3) that $In(K) = (n, m, 0)$. ∎

We now show that eventually μ can be reduced at an arbitrarily fast rate. We assume for simplicity that the QP (4.9.1) has only lower bounds and that these are zero. In the following lemma, $x(\mu)$ denotes the current iterate, μ the current barrier parameter, and μ_N the next barrier parameter to be used ($\mu_N < \mu$).

LEMMA 4.11.4. *Let x^* be a strong local minimum of QP at which we have strict complementary slackness. Let $x(\mu)$ be such that*

$$In(K_B(x(\mu),\mu)) = (n,\ m,\ 0),$$

$$\lim_{\mu\to 0} \|x(\mu) - x^*(\mu)\| = 0$$

and

$$\lim_{\mu\to 0} \|x^* - x^*(\mu)\| = 0,$$

where

$$K_B(x(\mu),\mu) = \begin{pmatrix} H_B(x(\mu),\mu) & A^T \\ A & 0 \end{pmatrix}.$$

If $\mu_N = \beta(\mu)\mu$, where $\beta(\mu)$ is the smallest positive scalar such that $In(K_B(x(\mu),\mu_N)) = (n,\ m,\ 0)$, then $\lim_{\mu\to 0}\beta(\mu) = 0$.

Proof. There is no loss of generality if we assume $x^* = (x_{FX}^*, x_{FR}^*)$, where x_{FX}^* denotes the variables on a bound and x_{FR}^* denotes the variables not on a bound. We shall also use the suffices to denote a partition of H and A, as in the preceding lemma.

It follows from the assumptions on x^* and $x(\mu)$ that

$$\lim_{\mu\to 0} \mu/(x_{FR}(\mu))_j^2 = 0 \tag{4.11.5}$$

and

$$\lim_{\mu \to 0} \mu/(x_{FX}(\mu))_j^2 = \infty.$$

Consequently, there exists $\epsilon > 0$ such that if $\mu < \epsilon$ then $\mu/(x_{FR}(\mu))_j^2 > \theta$ for all j, where $\theta < \infty$.

It follows from Lemma 4.11.3 and the assumptions on x^* that there exists γ such that $In(K_\gamma) = (n, m, 0)$, where

$$K_\gamma = \begin{pmatrix} H + \gamma I_{FX} & A^T \\ A & 0 \end{pmatrix}.$$

By definition,

$$K_B(x(\mu), \mu_N) = \begin{pmatrix} H + \beta(\mu)\mu X_{FX}^{-2} & \bar{H}^T & A_{FX}^T \\ \bar{H} & H_{FR} + \beta(\mu)\mu X_{FR}^{-2} & A_{FR}^T \\ A_{FX} & A_{FR} & 0 \end{pmatrix},$$

where $X_{FX} = \text{diag}(x_{FX}(\mu))$ and $X_{FR} = \text{diag}(x_{FR}(\mu))$. It follows from (4.11.5) that $In(\bar{K}) = (n_{FR}, m, 0)$, where

$$\bar{K} = \begin{pmatrix} H_{FR} + \beta(\mu) X_{FR}^{-2} & A_{FR}^T \\ A_{FR} & 0 \end{pmatrix}.$$

Therefore, provided $\beta(\mu)\mu/x_{FX}(\mu)_j^2 > \gamma$ for all j,

$$In(K_B(x(\mu), \mu_N)) = In(K_\gamma) = (n, m, 0).$$

Since $\lim_{\mu \to 0} \mu/x_{FX}(\mu)_j^2 = 0$ for all j, we have $\lim_{\mu \to 0} \beta(\mu) = 0$.

∎

4.12 A Simple Example

Consider solving the following QP by BARALG:

$$\begin{aligned} \underset{x,y \in \mathcal{R}}{\text{minimize}} \quad & Q(x,y) = (x + y - 3)^2 - (x - y - 2)^2 \\ \text{subject to} \quad & \\ & 3x + y - 1 \geq 0 \\ & x - y + 1 \geq 0 \\ & -x - y + 5 \geq 0 \\ & -x + 3y + 4 \geq 0 \\ & x \geq 0 \\ & y \geq 0. \end{aligned}$$

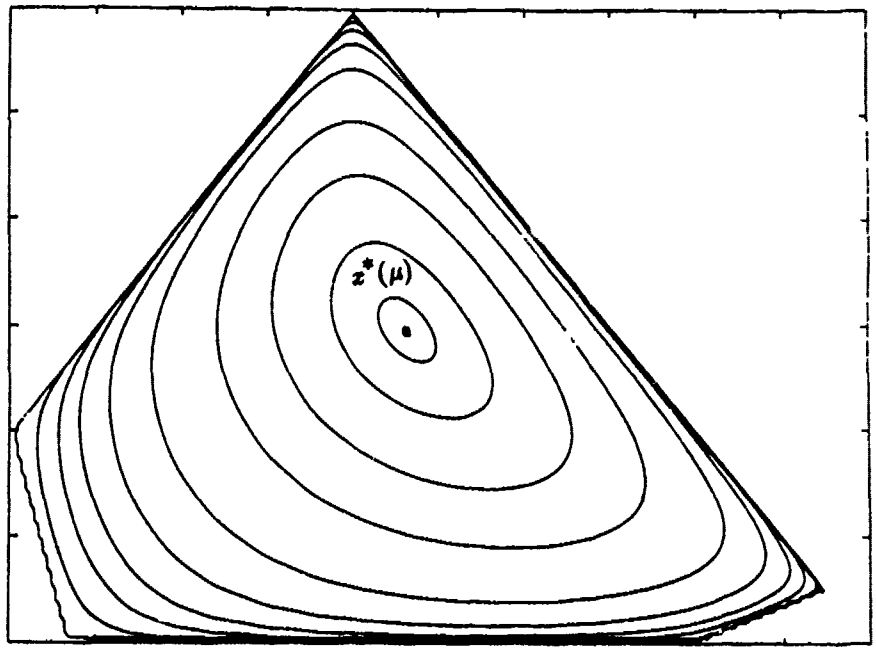

Figure 4.1: Contours of $B(x, \mu)$ for $\mu = 10$

Figures 4.1–4.3 show contours of the logarithmic barrier function for decreasing values of the barrier parameter: $\mu = 10, 1, 0.1$. Clearly for $\mu = 10$ the barrier function is convex. Although the barrier function is not convex for $\mu = 1$ the minimizer is the only stationary point. It follows that the first subproblem could have been terminated at any stage. Even though for $\mu = 0.1$ the function now has more than one stationary point, the required minimizer could be found provided the initial point was a quite modest approximation to the minimizer for the second subproblem. Had the reduction in μ been less severe, an even cruder approximation to the minimizer of the second subproblem would suffice.

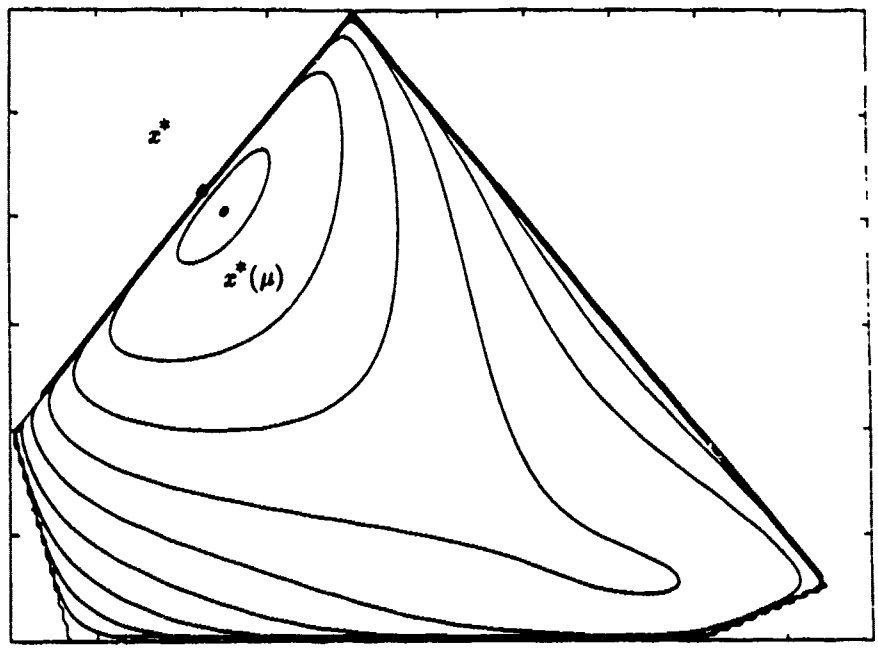

Figure 4.2: Contours of $B(x, \mu)$ for $\mu = 1$

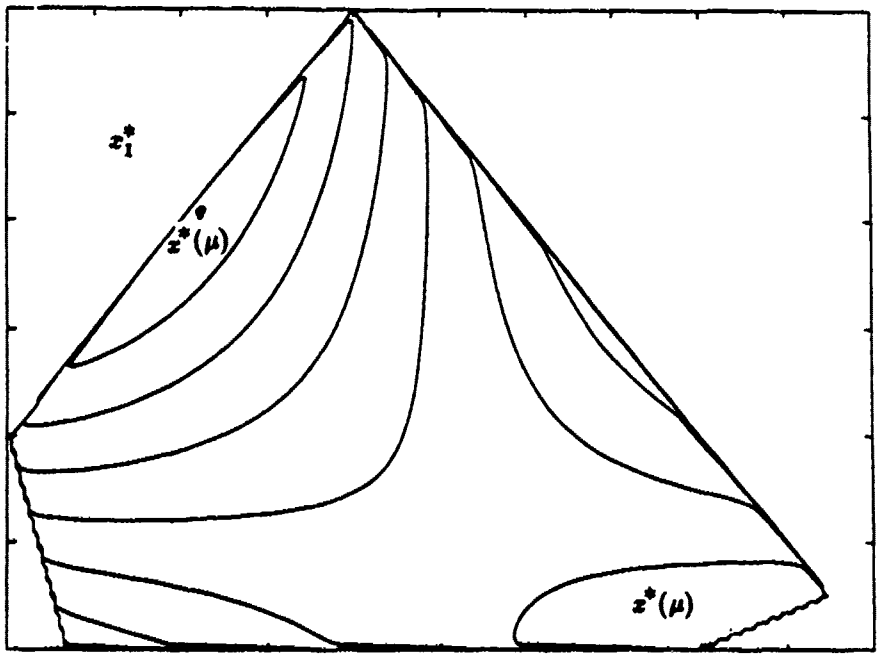

Figure 4.3: Contours of $B(x, \mu)$ for $\mu = 0.1$

Chapter 5

Sensitivity Analysis

In Chapters 3 and 4 we have described a model barrier algorithm that theoretically converges to the solution of a QP. To prove that the algorithm is numerically viable, we must show that the solution to the subproblems can be computed in a stable fashion. To this end, we perform a detailed sensitivity analysis of the KKT system arising from barrier methods. A key result of this analysis is that the ill-conditioning normally associated with the Hessian in barrier methods is benign when the logarithmic terms are applied only to simple bounds. Thus, a stable barrier algorithm is realized.

5.1 Barrier Methods Applied to General IQP Problems

In general, unconstrained minimization techniques are not well suited for minimizing barrier functions (see [Mur71a]). This is primarily due to the progressive ill-conditioning of the Hessians as the solution is approached. In this section we briefly illustrate the structure of the Hessian and explain why its condition number is unbounded. In later sections we prove that accurate subproblem solutions can be obtained despite this ill-conditioning when the barrier terms are applied only to simple bounds.

Consider the general IQP problem,

$$\begin{array}{lll} IQP & \underset{x\in\Re^n}{\text{minimize}} & F(x) = c^Tx + \frac{1}{2}x^THx \\ & \text{subject to} & \mathcal{A}x \geq \beta, \end{array} \tag{5.1.1}$$

where the Hessian H is an $n \times n$ symmetric matrix, $\mathcal{A}$ is an $m \times n$ matrix, and c and β are vectors of dimension n and m respectively. Let x^* denote the minimizer and let A be the $\bar{m} \times n$ matrix associated with the constraints that are active at x^*.

Applying a logarithmic barrier function to the inequality yields the following unconstrained minimization problem:

$$\underset{x\in\Re^n}{\text{minimize}} \quad B(x,\mu) = c^Tx + \frac{1}{2}x^THx - \mu\sum_{i=1}^{m}\ln(a_i^Tx - \beta_i), \tag{5.1.2}$$

where a_i^T refers to the i-th row of $\mathcal{A}$. By definition of an unconstrained minimizer,

the following relation holds when $x = x^*(\mu)$:

$$\nabla B = c + Hx - \mu \sum_{i=1}^{m} \frac{a_i}{r_i} = g - \mathcal{A}^T \begin{bmatrix} \mu/r_1 \\ \cdot \\ \cdot \\ \cdot \\ \mu/r_m \end{bmatrix} = 0, \tag{5.1.3}$$

where $r_i = a_i^T x - \beta_i$. Thus, the gradient of $F(x)$ at $x^*(\mu)$ is a nonnegative linear combination of all the constraints. Further, if the constraint $a_i^T x \geq \beta_i$ (of the original problem) is not active at x^*, its corresponding coefficient μ/r_i must approach zero (because r_i is bounded away from zero in a neighborhood of x^*). On the other hand, if the constraint is active at x^*, it can be shown that its coefficient approaches the corresponding Lagrange multiplier of the original problem (assuming A has full row rank). That is, for sufficiently small μ it follows from (5.1.3) that

$$\frac{\mu}{a_i^T x^*(\mu) - b_i} = \pi_i^* + O(\mu).$$

To arrive at the Hessian of the barrier function, we differentiate (5.1.3):

$$H_B = H + \mathcal{A}^T \begin{bmatrix} \mu/r_1^2 & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \mu/r_m^2 \end{bmatrix} \mathcal{A}.$$

We shall focus on the values of H_B at $x^*(\mu)$ as $\mu \to 0$.

LEMMA 5.1.1. *Let H_B be the Hessian of the barrier function (5.1.2) evaluated at $x^*(\mu)$. Let A be the $\bar{m} \times n$ submatrix of $\mathcal{A}$ associated with the constraints active at x^*, and assume A has full row rank $\bar{m}$. If $0 < \bar{m} < n$, as the solution of the original problem (5.1.1) is approached, the condition number of the Hessian becomes unbounded. Further, as $\mu \to 0$, the Hessian has $\bar{m}$ unbounded eigenvalues with eigenvectors in the range of A^T and $(n - \bar{m})$ bounded eigenvalues with eigenvectors in the null space of A.*

Proof. For inactive constraints, r_i is bounded away from zero and hence μ/r_i and μ/r_i^2 approach zero as $x^*(\mu)$ approaches x^*. For active constraints, the quantity μ/r_i approaches the Lagrange multiplier. Thus, $\bar{m}$ ratios μ/r_i^2 must be unbounded as $r_i \to 0$. This implies that if $0 < \bar{m} < n$, the dominant rank-deficient matrix causes the condition number of the Hessian to become unbounded and the result follows.

∎

Finally, we remark that the ill-conditioning of the Hessian matrices of barrier function does not result from the barrier parameter, but from the singularities caused by the active constraints. (See [Mur71b] for more details.)

5.2 Standard Formulation of QP Problems

In this thesis, we only consider the barrier method applied to QP problems in standard form:

$$\begin{array}{ll} \underset{x\in\Re^n}{\text{minimize}} & Q(x) \equiv c^T x + \frac{1}{2}x^T H x \\ \text{subject to} & Ax = b, \quad l \leq x \leq u. \end{array} \tag{5.2.1}$$

As we illustrate in this chapter, the standard formulation is of fundamental importance to the successful application of the barrier method. An intuitive understanding can be obtained by examining the Hessian of the barrier function. For (5.2.1) the barrier function is given by (4.9.2) and the corresponding Hessian is given by (4.9.4):

$$H_B = H + \mu D_B. \tag{5.2.2}$$

Notice that in this special case the effects of the logarithmic terms appear only on the diagonal of the Hessian. The $\bar{m}$ singularities of Lemma 5.1.1 are therefore separable. It is in fact this decoupling that allows accurate solutions to be computed. In the sensitivity analysis that follows, we make specific use of this particular form of the Hessian.

Let $Kv = w$ denote the KKT system associated with the barrier subproblem arising from the standard formulation (5.2.1).

We have said H_B is ill-conditioned. It is clear from the nature of the ill-conditioning of H_B that K is also ill-conditioned. We show that v can be computed reliably nevertheless.

5.3 Standard Sensitivity Analysis

Given the KKT system $Kv = w$, our aim is to examine how perturbations in K and w affect the solution v. In this section we recall some results from the standard perturbation theory for solving linear systems.

Let δK and δw be perturbations for K and w respectively, so that the perturbed KKT system can be expressed as

$$(K + \delta K)(v + \delta v) = (w + \delta w), \tag{5.3.1}$$

which we shall write as

$$\widetilde{K}\tilde{v} = \tilde{w}. \tag{5.3.2}$$

Even when K is nonsingular, $\widetilde{K}$ may be singular if δK is not restricted. It is known [GL89] that if δK is such that $\|\delta K\| \leq 1/\|K^{-1}\|$, then the following hold:

- $K + \delta K$ is nonsingular;
- $(K + \delta K)^{-1} = (I + K^{-1}\delta K)^{-1}K^{-1} = \sum_{i=0}^{\infty}(K^{-1}\delta K)^i K^{-1}$;
- The relative error in the solution of (5.3.1) can be bounded by

$$\frac{\|\delta v\|}{\|v\|} \leq \frac{\text{cond}(K)}{1 - \text{cond}(K)\rho(K)}\left(\rho(K) + \rho(w)\right), \tag{5.3.3}$$

where $\text{cond}(K) = \|K^{-1}\|\|K\|$. The quantities $\rho(K) = \|\delta K\|/\|K\|$ and $\rho(w) = \|\delta w\|/\|w\|$ constitute the relative changes in K and w respectively.

The condition number $\text{cond}(K)$ attempts to measure the worst possible effect on the solution v when K and w are perturbed by a small amount.

Unfortunately, as we pointed out in the previous section, for the linear systems in consideration, $\text{cond}(K) \to \infty$ as $\mu \to 0$. Ironically the system becomes increasingly ill-conditioned precisely when we are more interested in solving it accurately, that is, as we approach the solution.

5.4 Useful Identities and Inequalities

$$\begin{aligned}
A^{-1} - B^{-1} &= A^{-1}(B - A)B^{-1} \\
(I - A)^{-1} &= \sum_{i=0}^{\infty} A^i \\
(A + \gamma D)^{-1} &= \frac{1}{\gamma}D^{-1}\sum_{i=0}^{\infty}(-1)^i\gamma^{-i}(AD^{-1})^i \\
(A + \gamma D)^{-1} &= \gamma^{-1}D^{-1} - \gamma^{-2}D^{-1}AD^{-1} + O(\gamma^{-3})
\end{aligned}$$

5.5 Analysis of the KKT System

The KKT system for the barrier subproblem is

$$\begin{pmatrix} H_B & A^T \\ A & \end{pmatrix}\begin{pmatrix} -p \\ q \end{pmatrix} = \begin{pmatrix} g_L \\ r \end{pmatrix}, \quad K \equiv \begin{pmatrix} H_B & A^T \\ A & \end{pmatrix}, \tag{5.5.1}$$

where $H_B = H + \mu D_B$ and $D_B = D_\ell^2 + D_u^2$ (see Equation (4.9.3)). Without loss of generality and for clarity of exposition, we will consider only problems with nonnegativity bounds, so that for the remainder of this chapter, $D_B = \text{diag}(1/x_j^2)$.

We are interested in analyzing the behavior of the solution of system (5.5.1), especially the limiting behavior as the barrier parameter tends to zero. For the purpose of the analysis we will consider a particular ordering of the KKT equations. The

motivation is to distinguish (or treat separately) those elements of D_B that become unbounded as we approach the solution.[1] Let us assume that the rows and columns of D_B have been permuted so that the unbounded elements appear first. If we define

$$\begin{aligned} \beta &= x_s^2 = \min_j x_j^2 \\ \gamma &= \mu/\beta \\ d_j &= \beta/x_j^2 \\ D &= \text{diag}(d_j), \end{aligned}$$

then

- $H_B = H + \mu D_B = H + \gamma D$;
- $0 \leq d_j \leq 1$, and $\|D\| = 1$.

This ordering induces the following partition in the matrices D, H_B and K:

$$D = \begin{pmatrix} D_1 & \\ & D_2 \end{pmatrix}, \quad H_B = H + \gamma \begin{pmatrix} D_1 & \\ & D_2 \end{pmatrix},$$

$$K(\gamma) = \begin{pmatrix} H_1 + \gamma D_1 & H_{21}^T & A_1^T \\ H_{21} & H_2 + \gamma D_2 & A_2^T \\ A_1 & A_2 & \end{pmatrix},$$

where $\gamma D_1 \to \infty$ as $\mu \to 0$, in contrast to γD_2, which by construction remains bounded as $\mu \to 0$.

Without loss of generality we can assume that the original QP has been scaled so that $\|H\| \approx 1$ and $\|A\| \approx 1$. Therefore, the system of interest (5.5.1) becomes

$$K(\gamma)v(\gamma) = \begin{pmatrix} H + \gamma D & A^T \\ A & 0 \end{pmatrix} v(\gamma) = w, \tag{5.5.2}$$

where γ is a positive scalar; K and D are square matrices; K is symmetric, indefinite and $\| K \| \approx 1$; D is diagonal, positive semi-definite and $\| D \| = 1$. With the chosen partitioning we also have $\|D_1\| = 1$ (since x_s will be in D_1). We are interested in the quantity

$$\lim_{\gamma \to \infty} v(\gamma). \tag{5.5.3}$$

[1] If there are no unbounded elements in D_B, i.e., if none of the components of x approaches a bound, standard sensitivity analysis applies.

Equation (5.5.2) is equivalent to

$$\begin{pmatrix} K_1 & K_3^T \\ K_3 & K_2 \end{pmatrix} \begin{pmatrix} v_1(\gamma) \\ v_2(\gamma) \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}, \tag{5.5.4}$$

where

$$K_1 = H_1 + \gamma D_1,$$

$$K_2 = \begin{pmatrix} H_2 + \gamma D_2 & A_2^T \\ A_2 & 0 \end{pmatrix},$$

$$K_3 = \begin{pmatrix} H_{21} \\ A_1 \end{pmatrix}.$$

Note that K_2 is a KKT system of smaller dimension, namely the submatrix obtained from the original system after eliminating variables that are near their bounds.

Let $S \equiv K/K_1$ be the Schur complement of K with respect to K_1. The system (5.5.4) can be written as

$$K_1 v_1(\gamma) = w_1 - K_3^T v_2(\gamma), \tag{5.5.5}$$

$$S v_2(\gamma) = w_2 - K_3 K_1^{-1} w_1. \tag{5.5.6}$$

In order to find an expression for (5.5.3), we need to present three lemmas.

LEMMA 5.5.1. *If γ is sufficiently large, the matrix K_1 is nonsingular and its inverse has the form*

$$K_1^{-1} = \frac{1}{\gamma} D_1^{-1} - \frac{1}{\gamma^2} D_1^{-1} H_1 D_1^{-1} + O(\gamma^{-3}). \tag{5.5.7}$$

Proof:

$$\begin{aligned} K_1 &= H_1 + \gamma D_1 \\ &= (I + \gamma^{-1} H_1 D_1^{-1}) \gamma D_1. \end{aligned}$$

Hence,

$$K_1^{-1} = \gamma^{-1} D_1^{-1} (I + \gamma^{-1} H_1 D_1^{-1})^{-1}.$$

If γ is sufficiently large we have

$$(I + \gamma^{-1} H_1 D_1^{-1})^{-1} = I - \gamma^{-1} H_1 D_1^{-1} + O(\gamma^{-2}).$$

The required result follows.

∎

LEMMA 5.5.2. *If K_2 is nonsingular and γ is sufficiently large, S is nonsingular and its inverse can be written as*

$$S^{-1} = K_2^{-1} + \frac{1}{\gamma} K_2^{-1} K_3 D_1^{-1} K_3^T K_2^{-1} + O(\gamma^{-2}).$$

Proof: We have from (4.11.4) in Lemma 4.11.3 that if γ is large enough,

$$In(S) = In(K_2).$$

It follows that for γ large enough, S is nonsingular. In all that follows we assume that γ is always chosen suitably large.

By definition we have

$$\begin{aligned} S^{-1} &= (K_2 - A(\gamma))^{-1} \\ &= K_2^{-1}(I - A(\gamma)K_2^{-1})^{-1} \\ &= K_2^{-1} + K_2^{-1}A(\gamma)K_2^{-1} + \sum_{i=2}^{\infty} K_2^{-1}[A(\gamma)K_2^{-1}]^i, \end{aligned} \tag{5.5.8}$$

where $A(\gamma) = K_3 K_1^{-1} K_3^T$. It follows from Lemma 5.5.1 and the definition of $A(\gamma)$ that

$$\begin{aligned} A(\gamma) &= K_3[\gamma^{-1}D_1^{-1} - \gamma^{-2}D_1^{-1}H_1D_1^{-1} + O(\gamma^{-3})]K_3^T \\ &= \frac{1}{\gamma} K_3 D_1^{-1} K_3^T + O(\gamma^{-2}). \end{aligned} \tag{5.5.9}$$

The required results follows from substituting this expression for $A(\gamma)$ into (5.5.8). ∎

LEMMA 5.5.3. *If K_2 and D_1 are nonsingular and $\gamma > \|D_1^{-1}H_1\|$, then*

$$\begin{aligned} v_2(\gamma) &= K_2^{-1}w_2 + \gamma^{-1}K_2^{-1}K_3D_1^{-1}(K_3^TK_2^{-1}w_2 - w_1) + O(\gamma^{-2}), \\ v_1(\gamma) &= \frac{1}{\gamma} D_1^{-1}(w_1 - K_3^TK_2^{-1}w_2) + O(\gamma^{-2}). \end{aligned}$$

Proof: From (5.5.6) and Lemma 5.5.2 we have

$$\begin{aligned} v_2(\gamma) &= S^{-1}(w_2 - K_3K_1^{-1}w_1) \\ &= (K_2^{-1} + \frac{1}{\gamma} K_2^{-1}K_3D_1^{-1}K_3^TK_2^{-1})(w_2 + K_3K_1^{-1}w_1) + O(\gamma^{-2}). \end{aligned}$$

Substituting for K_1^{-1} from Lemma 5.5.1 gives the first required result.

Similarly, from (5.5.5) and Lemma 5.5.1 we have

$$\begin{aligned} v_1(\gamma) &= K_1^{-1}(w_1 - K_3^T v_2(\gamma)) \\ &= \frac{1}{\gamma} D_1^{-1}(w_1 - K_3^T v_2(\gamma)) + O(\gamma^{-2}). \end{aligned}$$

Substituting for $v_2(\gamma)$ gives the second required result.

∎

From the last lemma we can derive the limiting value of $v(\gamma)$:

$$\lim_{\gamma \to \infty} v_1(\gamma) = 0,$$

$$\lim_{\gamma \to \infty} v_2(\gamma) = K_2^{-1} w_2.$$

These results indicate that when $\gamma \to \infty$ (or equivalently when $\mu \to 0$) the KKT system decouples. This is precisely what we would obtain if we fixed the components of v that are on a bound at the solution and solve for the reduced system.

5.6 Specific Sensitivity Analysis

Here we analyze the sensitivity of the solution of the KKT systems arising in barrier methods. It was shown in the previous section that these systems can be expressed as

$$\begin{pmatrix} H_1 + \gamma D_1 & K_3^T \\ K_3 & K_2 \end{pmatrix} \begin{pmatrix} v_1(\gamma) \\ v_2(\gamma) \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}, \qquad K \equiv \begin{pmatrix} K_1 & K_3^T \\ K_3 & K_2 \end{pmatrix}, \tag{5.6.1}$$

where $K_1 = H_1 + \gamma D_1$.

In the following analysis it is tacitly assumed that $\|v(\gamma)\| > 0$ for all γ sufficiently large. We wish to address how sensitive the solution of (5.6.1) is to perturbations in the matrix K and the vector w. Thus if

$$(K + \delta K)(v + \delta v) = (w + \delta w), \tag{5.6.2}$$

where

$$\delta K = \begin{pmatrix} \delta H_1 + \gamma \delta D_1 & \delta K_3^T \\ \delta K_3 & \delta K_2 \end{pmatrix} \quad \text{and} \quad \delta v = \begin{pmatrix} \delta v_1 \\ \delta v_2 \end{pmatrix},$$

we are interested in how the relative and absolute perturbations behave when $\gamma \to \infty$. Since $\lim_{\gamma \to \infty} \|v_1(\gamma)\| = \lim_{\gamma \to \infty} \|v_1(\gamma) + \delta v_1(\gamma)\| = 0$, our interest is in the behavior of the perturbation δv_2.

We use the Schur complement, so that for the original KKT system we have

$$\begin{aligned} K_1 v_1 &= w_1 - K_3^T v_2, \\ S v_2 &= w_2 - K_3 K_1^{-1} w_1, \end{aligned} \tag{5.6.3}$$

and for the perturbed KKT system

$$\begin{aligned} \widetilde{K}_1 \tilde{v}_1 &= \tilde{w}_1 - \widetilde{K}_3^T \tilde{v}_2, \\ \tilde{S} \tilde{v}_2 &= \tilde{w}_2 - \widetilde{K}_3 \widetilde{K}_1^{-1} \tilde{w}_1, \end{aligned} \tag{5.6.4}$$

where

$$\begin{aligned} \tilde{v}_1 &= v_1 + \delta v_1, \\ \tilde{v}_2 &= v_2 + \delta v_2, \\ \widetilde{K}_1 &= K_1 + \delta H_1 + \gamma \delta D_1, \\ \widetilde{K}_2 &= K_2 + \delta K_2, \\ \widetilde{K}_3 &= K_3 + \delta K_3, \\ \tilde{S} &= \widetilde{K}_2 - \widetilde{K}_3 \widetilde{K}_1^{-1} \widetilde{K}_3^T. \end{aligned}$$

It follows from Lemma 5.5.2 and the nonsingularity of K_2 (a consequence of the nonsingularity of K) that

$$\lim_{\gamma \to \infty} S = K_2,$$

$$\lim_{\gamma \to \infty} \tilde{S} = K_2 + \delta K_2.$$

Also from Lemma 5.5.1 we obtain

$$\lim_{\gamma \to \infty} \widetilde{K}_1^{-1} = \lim_{\gamma \to \infty} K_1^{-1} = 0.$$

In the limit (allowing $\gamma \to \infty$), we therefore have from (5.6.3) and (5.6.4) that

$$\begin{aligned} K_2 v_2 &= w_2, \\ (K_2 + \delta K_2)(v_2 + \delta v_2) &= w_2 + \delta w_2. \end{aligned}$$

We can now apply the classical analysis of perturbed systems to obtain

$$\frac{\|\delta v_2\|}{\|v_2\|} \leq \frac{\mathrm{cond}(K_2)}{1 - \mathrm{cond}(K_2)\rho(K_2)} (\rho(K_2) + \rho(w_2)).$$

These results show that the sensitivity of systems involving K becomes the same as the sensitivity of the systems involving K_2 as $\gamma \to \infty$.

5.7 An Example

The analysis demonstrates that provided the problem is in standard form, the sensitivity of the solution to the KKT system arising from the barrier subproblem depends in the limit on the condition of the underlying KKT system for the optimal active set for the original problem and not on the condition of K. Here we demonstrate by a counterexample that this is not the case for problems in nonstandard form. We contrast the relative errors made when solving two specific systems of equations; one of the type arising from problems in nonstandard form and the other of the type arising from problems in standard form.

Suppose the QP problem has some general inequalities, i.e.,

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}}\ & Q(x) = c^Tx + \tfrac{1}{2}x^THx \\ \text{subject to}\ & Ax = b, \\ & \bar{A}x \geq \bar{b}. \end{aligned} \tag{5.7.1}$$

When a barrier transformation is applied, the subproblem is of the form

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}}\ & B(x,\mu) = c^Tx + \tfrac{1}{2}x^THx - \mu\sum_i \ln r_i(x) \\ \text{subject to}\ & Ax = b, \end{aligned} \tag{5.7.2}$$

where $r_i(x) = \bar{a}_i^Tx - \bar{b}_i$.

Superficially this subproblem looks identical to the subproblem for the standard-form QP (4.9.2). The key difference is the form of the Hessian of the two barrier functions. In the standard-form case, H_B has the form

$$H_B = H + \gamma D,$$

where D is a diagonal matrix satisfying $0 < \|D\| < \infty$. In the nonstandard form the Hessian is of the form

$$H_B = H + \gamma\bar{A}^T\bar{D}\bar{A},$$

where $\bar{D}$ is also a diagonal matrix. In both cases we have

$$\lim_{x\to x^*} \gamma = \infty.$$

In the standard-form case there is a loss of figures in some of the diagonal elements of H when H_B is computed in finite precision. The previous analysis shows that this is not of significance. For the nonstandard form there is a loss of figures in all the elements of H for which $\bar{A}^T\bar{A}$ has a nonzero element (which could be all elements of H). Eventually, loss of figures in the off-diagonal elements of H becomes catastrophic.

γ	e_B
1	1.5e-15
10^5	2.4e-11
10^{10}	1.4e-6
10^{15}	2.1e-1

Table 5.1: Relative error for systems arising from problems in nonstandard form.

To illustrate, we first consider systems of the form

$$(H + \gamma B^T B)x = d, \tag{5.7.3}$$

where H is an $n \times n$ symmetric orthogonal matrix and B is an $m \times n$ matrix with $m < n$. The matrix B (like H) has a condition number of one, so that errors in the computed solution arise from the *form* of equation (5.7.3). For $\gamma \gg 1$, the condition number of $H + \gamma B^T B$ is approximately equal to γ. The specific choices of H, B and d are

$$H = \begin{pmatrix} -0.4179 & 0.4528 & 0.2045 & 0.7606 \\ 0.4528 & 0.8554 & -0.0653 & -0.2429 \\ 0.2045 & -0.0653 & 0.9705 & -0.1097 \\ 0.7606 & -0.2429 & -0.1097 & 0.5920 \end{pmatrix},$$

$$B = \begin{pmatrix} -0.3844 & -0.4856 & 0.2404 & 0.7474 \\ 0.6325 & 0.3162 & -0.3162 & 0.6325 \end{pmatrix} \quad \text{and} \quad d = \begin{pmatrix} 4.3 \\ 1.3 \\ -1.7 \\ 0.5 \end{pmatrix}.$$

We solved (5.7.3) for a variety of values of γ, using MATLAB [MLB87] on a DEC VAX workstation with approximately 16 decimals of precision. Let $\tilde{x}$ denote the computed solution and $e_B = \|\tilde{x} - x\|/\|x\|$ its relative error.

It is clear from Table 5.1 that the loss of figures in the computed solution to systems of the form (5.7.3) is close to the bound predicted by the standard sensitivity analysis.

We now consider equations of the form

$$(H + \gamma D)x = d, \tag{5.7.4}$$

γ	e_D
1	1.6e-15
10^5	6.3e-17
10^{10}	1.5e-25
10^{15}	1.4e-16

Table 5.2: Relative error for systems arising from problems in standard form.

where D is a diagonal matrix of rank $m < n$. Specifically we use the same matrix H of the previous example and choose $D = \text{diag}(1, 1, 0, 0)$. Let $\tilde{x}$ denote the computed solution and $e_D = \|\tilde{x} - x\|/\|x\|$ its relative error.

According to the special sensitivity analysis, the loss of figures in the solution should not depend on the condition number of $H + \gamma D$ but on the condition of

$$H_2 = \begin{pmatrix} 0.9705 & -0.1097 \\ -0.1097 & 0.5920 \end{pmatrix}$$

as $\gamma \to \infty$. This matrix has a condition number of only 1.8. It can be seen from Table 5.2 that as predicted the errors are indeed very small.

Chapter 6

Special QP Problems

6.1 Introduction

In this chapter we discuss various special cases of QP. Specifically we consider the case when Z^THZ is positive semidefinite and various special forms of H. Much of what follows in this chapter is also applicable to the general case when the circumstances considered hold locally. A key property when Z^THZ is positive semidefinite is that the minimizer of the barrier subproblem is unique. It follows that there exists a single constrained stationary point. Consequently we may apply Newton's method directly to the nonlinear equations that define the stationary point.

It is often possible to take advantage of specific structure in H when solving the KKT system of equations. We are particularly interested in the case when H is trivially invertible. The last special case considered is when the only constraints are simple bounds.

Throughout Sections 6.2–6.6 we assume that Z^THZ is positive semidefinite

6.2 A Primal Method

Let us recall the *primal barrier subproblem*

$$\begin{array}{ll} \underset{x\in\Re^n}{\text{minimize}} & B(x,\mu) = c^Tx + \frac{1}{2}x^THx - \mu\sum_{j=1}^{n}\ln x_j \\ \text{subject to} & Ax = b, \end{array} \tag{6.2.1}$$

and the associated KKT equations

$$\begin{pmatrix} H_B & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} \Delta x \\ -\Delta\pi \end{pmatrix} = -\begin{pmatrix} g_B - A^T\pi \\ Ax - b \end{pmatrix}. \tag{6.2.2}$$

If Z^THZ is positive semidefinite, the matrix Z^TH_BZ is positive definite and the barrier subproblem has a unique solution corresponding to a unique constrained stationary point.

The equations defining a constrained stationary point of (6.2.1) are:

$$\begin{aligned} c + Hx - \mu X^{-1}e - A^T\pi &= 0 \\ Ax - b &= 0. \end{aligned}$$

Defining $z = \mu X^{-1}e$ gives

$$f_p(z, x, \pi) \equiv \begin{pmatrix} z - \mu X^{-1}e \\ c + Hx - z - A^T\pi \\ Ax - b \end{pmatrix} = 0. \tag{6.2.3}$$

The vector v will denote the vector $(z, x, -\pi)$. The Newton direction $\Delta v = (\Delta z,\ \Delta x,\ -\Delta\pi)$ satisfies the linear system

$$J_p \Delta v = -f_p, \tag{6.2.4}$$

where

$$J_p = \begin{pmatrix} I & \mu X^{-2} & 0 \\ -I & H & A^T \\ 0 & A & 0 \end{pmatrix}.$$

Apart from the last block of columns being multiplied by -1, J_p is the Jacobian of f_p. We shall refer to J_p as the Jacobian. For the Jacobian to be well defined we assume $x > 0$. It is easily seen that if a step α were taken such that any $x_j + \alpha\Delta x_j = 0$ for some j, then $\|f_p\|^2 = f_p^T f_p = \infty$. Therefore there exists a minimizer of

$$\underset{\beta}{\text{minimize}} \quad \{\|f_p(v + \beta\Delta v)\|^2 \quad | \quad \beta > 0\},$$

say $\bar{\beta}$, such that $x + \bar{\beta}\Delta x > 0$ if $x > 0$.

The next iterate v_{k+1} is given by $v_{k+1} = v_k + \alpha\Delta v$, where $v_k = (z_k, x_k, -\pi_k)$ and α is chosen to ensure a sufficient decrease in $\|f_p\|^2$. Specifically we choose α to satisfy the Goldstein-Armijo conditions:

$$0 < -2\gamma_1\alpha\Delta v^T J_p^T f_p(v_k) \le \|f_p(v_k)\|^2 - \|f_p(v_{k+1})\|^2 \le -2\gamma_2\alpha\Delta v^T J_p^T f_p(v_k),$$

where γ_1 and γ_2 are scalars satisfying $0 < \gamma_1 \le \gamma_2 < 1$. Since $J_p\Delta v = -f_p(v_k)$ the conditions reduce to

$$0 < 2\gamma_1\alpha \le 1 - \frac{\|f_p(v_{k+1})\|^2}{\|f_p(v_k)\|^2} \le 2\gamma_2\alpha.$$

The trial values of the steplength are defined in terms of an initial step $\alpha^{(0)} = \min(1,\ \theta\alpha_m)$ and a positive scalar w ($w < 1$), where α_m is the largest feasible step

along Δv and $0 < \theta < 1$. Typically $\theta = .99$ and $w = 0.5$. The value of α is taken as the first member of the sequence $\{w^j \alpha^{(0)}\}$, $j = 0, 1, \ldots$, for which the conditions are satisfied. It is because $\bar{\beta}$ exists that we can assert that there are suitable points satisfying the Goldstein-Armijo conditions.

Given that the initial value of v, say v_0, satisfies $\|f_p(v_0)\| < \infty$, it follows that there exists $M < \infty$ such that $(x_k)_j > M/\mu$ for all k. Consequently, $\|J_p\|$ is bounded and J_p has a bounded condition number at all iterates. Hence Newton's method is well defined. It follows from standard convergence analysis that $\lim_{k\to\infty} v_k = (z^*(\mu), x^*(\mu), -\pi^*(\mu))$.

The vectors Δx and $\Delta\pi$ are identical to those defined by the KKT equations (6.2.2). Indeed if Δz is eliminated from (6.2.4) we also eliminate z, giving the KKT equations

$$\begin{pmatrix} H + \mu X^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta\pi \end{pmatrix} = - \begin{pmatrix} c + Hx - \mu X^{-1} e - A^T\pi \\ Ax - b \end{pmatrix}. \tag{6.2.5}$$

Newton's method applied to the nonlinear equations (6.2.3) differs from the preceding definition of Newton's method (applied to the barrier subproblem) in that Δx and α are used to ensure a sufficient decrease in $\|f_p\|$ rather than $B(x, \mu)$. An advantage of the new formulation is that $\|f_p\|$ is easier to compute than $B(x, \mu)$.

By utilizing the Schur complement of $H + \mu X^{-2}$ in the KKT matrix we obtain equations in just $\Delta\pi$, namely

$$A(H + \mu X^{-2})^{-1} A^T \Delta\pi = b - Ax + A(H + \mu X^{-2})^{-1}(g_B - A^T\pi), \tag{6.2.6}$$

where $g_B = c + Hx - \mu X^{-1} e$. In contrast to the linear programming case, these equations do not appear to be an attractive option for computing $\Delta\pi$ unless H happens to be diagonal (cf. [CLMS90]). In Section 6.7 we show that H can be made diagonal for any semidefinite QP.

It is clear from (6.2.5) that Δx is independent of z. A closer inspection of (6.2.5) reveals that Δx is also independent of π. It is therefore unnecessary to take the same step in the x variable as that taken in z and π.

We could define the iteration as follows:

$$\begin{aligned} x_{k+1} &= x_k + \alpha \Delta x \\ z_{k+1} &= z_k + \alpha_z \Delta z \\ \pi_{k+1} &= \pi_k + \alpha_z \Delta\pi, \end{aligned} \tag{6.2.7}$$

where α is chosen as before, but α_z is chosen to ensure $z_{k+1} > 0$ if $z_k > 0$. The intent is to choose $z_0 > 0$, which then ensures $z_k > 0$ for all k.

In addition to the sequence $\{x_k\}$, we may recur the sequence $\{\bar{x}_k\}$ as follows:

$$\begin{aligned} \bar{x}_0 &= x_0 \\ \bar{x}_{k+1} &= \bar{x}_k + \alpha_z \Delta\bar{x}_k, \end{aligned}$$

where

$$\Delta\bar{x}_k = x_k + \Delta x_k - \bar{x}_k.$$

We could choose α_z to be the maximum feasible step (subject to it being less than unity). However, since we know at the required solution $z^* > 0$, we choose instead to take some step that ensures $z_{k+1} > 0$. We have yet to formulate the best rules for choosing α_z. Our purpose here is to note that we do have the choice. Our reason for choosing α_z different from α is that we are able to generate a dual feasible point and maintain such a point, as the following lemma proves.

LEMMA 6.2.1. *If at the K-th iteration we have*

$$\begin{aligned} c + H\bar{x}_K - z_K - A^T\pi_K &= 0 \\ A\bar{x}_K &= b, \end{aligned} \tag{6.2.8}$$

it follows that

$$\begin{aligned} c + H\bar{x}_k - z_k - A^T\pi_k &= 0 \\ A\bar{x}_k &= b \end{aligned}$$

for $k \geq K$.

Proof. It is enough to show that the result holds for $k = K + 1$. By definition we have

$$\begin{aligned} c + H(x_K + \Delta x_K) - (z_K + \Delta z_K) - A^T(\pi_K + \Delta\pi_K) &= 0 \\ A(x_K + \Delta x_K) &= b. \end{aligned}$$

It follows from these equations and the assumptions that

$$\begin{aligned} H(x_K + \Delta x_K - \bar{x}_K) - \Delta z_K - A^T\Delta\pi_K &= 0 \\ A(x_K + \Delta x_K - \bar{x}_K) &= 0. \end{aligned}$$

Since $\Delta\bar{x}_K = x_K + \Delta x_K - \bar{x}_K$ we have

$$\begin{aligned} H\Delta\bar{x}_K - \Delta z_K - A^T\Delta\pi_K &= 0 \\ A\Delta\bar{x}_K &= 0. \end{aligned}$$

It follows that

$$\begin{aligned} c + H\bar{x}_{K+1} - z_{K+1} - A^T\pi_{K+1} &= 0 \\ A\bar{x}_{K+1} &= b. \end{aligned}$$

An immediate consequence of this lemma is that if ever $\alpha_z = 1$, then all subsequent elements of the sequence $\{\bar{x}_k, z_k, \pi_k\}$ are dual feasible (even if $\alpha_z \neq \alpha$ at some stage). At such points we are able to compute a lower bound on the objective function at the solution. If ever $\alpha = 1$, then all subsequent elements of $\{x_k\}$ are primal feasible and we are able to compute an upper bound on the objective function at the solution. Once a primal and dual feasible solution are known, then at every subsequent iteration we may compute the duality gap given by

$$c^T x_k + \tfrac{1}{2} x_k^T H x_k - b^T \pi_k + \tfrac{1}{2} \bar{x}_k^T H \bar{x}_k,$$

which when $x_k = \bar{x}_k$ becomes

$$c^T x_k + x_k^T H x_k - b^T \pi_k.$$

Knowledge of the duality gap may be used to estimate a suitable change to the value of μ and as an overall termination criterion.

Since the sequence $\{x_k\}$ is unaffected by choosing $\alpha_z \neq \alpha$, and since a step bounded away from zero may still be taken, it follows that we still have $\lim_{k\to\infty} x_k = x^*(\mu)$. However, it is necessary to show that $\lim_{k\to\infty} z_k = z^*(\mu)$ and $\lim_{k\to\infty} \pi_k = \pi^*(\mu)$.

LEMMA 6.2.2. *If the sequences $\{z_k\}$ and $\{\pi_k\}$ are generated according to the above rules, then $\lim_{k\to\infty} z_k = z^*(\mu)$ and $\lim_{k\to\infty} \pi_k = \pi^*(\mu)$.*

Proof. By definition we have

$$\Delta z_k - \mu X_k^{-2} \Delta x_k = -z_k + \mu X_k^{-1} e.$$

Hence

$$z_k + \Delta z_k = \mu X_k^{-1} e + \mu X_k^{-2} \Delta x_k.$$

Since there exists ϵ such that $(x_k)_j > \epsilon > 0$, and since $\lim_{k\to\infty} \Delta x_k = 0$, it follows that

$$\lim_{k\to\infty} z_k + \Delta z_k = \lim_{k\to\infty} \mu X_k^{-1} e = z^*(\mu).$$

It also follows that there exists K such that $\alpha_z = 1$ for all $k \geq K$. Hence $\lim_{k\to\infty} z_k = z^*(\mu)$.

Since $\lim_{k\to\infty} \Delta x_k = 0$, there exists $\bar{K}$ such that $\alpha = \alpha_z = 1$ for all $k \geq \bar{K}$. Consequently, for $k \geq \bar{K}$ we have $\bar{x}_k = x_k$. It follows from this result and from Lemma 6.2.1 that for $k \geq \bar{K}$,

$$c + H x_k - z_k - A^T \pi_k = 0.$$

Since $\lim_{k\to\infty} z_k = z^*(\mu)$ and $\lim_{k\to\infty} x_k = x^*(\mu)$, it follows that $\lim_{k\to\infty} \pi_k = \pi^*(\mu)$. ∎

Since a unit step in all the variables is assured after a finite number of iterations, it follows that eventually both a primal and dual feasible point will be identified regardless of the value of μ.

6.3 A Dual Method

Let $z = c + Hx - A^T\pi$. The *dual barrier* subproblem can then be written as

$$\begin{aligned} \underset{x,\pi,z}{\text{minimize}} \quad & B(x,\pi,z,\mu) \equiv -b^T\pi + \tfrac{1}{2}x^THx - \mu\sum_{j=1}^{n}\ln z_j \\ \text{subject to} \quad & -z + c + Hx - A^T\pi = 0. \end{aligned} \tag{6.3.1}$$

A stationary point of (6.3.1) satisfies the equations

$$f_d(z,x,\pi) \equiv \begin{pmatrix} x - \mu Z^{-1}e \\ c + Hx - z - A^T\pi \\ Ax - b \end{pmatrix} = 0. \tag{6.3.2}$$

This system is equivalent to (6.2.3) in the sense that the solution to both systems with $x^*(\mu) > 0$ is identical. We could have arrived at (6.3.2) by multiplying the first n equations of f_p in (6.2.3) by XZ^{-1}. The Jacobian of f_d is given by

$$J_d = \begin{pmatrix} \mu Z^{-2} & I & 0 \\ -I & H & A^T \\ 0 & A & 0 \end{pmatrix}.$$

If we apply Newton's method to (6.3.2) then the following system is solved at each iteration:

$$J_d \begin{pmatrix} \Delta z \\ \Delta x \\ -\Delta\pi \end{pmatrix} = -\begin{pmatrix} x - \mu Z^{-1}e \\ c + Hx - z - A^T\pi \\ Ax - b \end{pmatrix}. \tag{6.3.3}$$

Observe that $\|f_d\| = \infty$ whenever an element of z is zero. The next iterate is given by taking a step α along $(\Delta z, \Delta x, \Delta\pi)$, and we now require z to remain positive. The proof that the sequence converges to the required solution follows an identical argument to that given for the primal method.

The corresponding KKT system is

$$\begin{pmatrix} H + \frac{1}{\mu}Z^2 & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta\pi \end{pmatrix} = -\begin{pmatrix} c + Hx - 2z - A^T\pi + \frac{1}{\mu}Z^2x \\ Ax - b \end{pmatrix}. \tag{6.3.4}$$

As in the primal case we can obtain equations in just $\Delta\pi$:

$$A(H+\frac{1}{\mu}Z^2)^{-1}A^T\Delta\pi = b-Ax+A(H+\frac{1}{\mu}Z^2)^{-1}(c+Hx-A^T\pi-2z+\frac{1}{\mu}Z^2x_j. \quad (6.3.5)$$

Since x appears linearly in (6.3.2) it follows that $\hat{x} = x + \Delta x$ is independent of x. Rearranging (6.3.4) gives

$$\begin{pmatrix} H+\frac{1}{\mu}Z^2 & A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} \hat{x} \\ -\Delta\pi \end{pmatrix} = -\begin{pmatrix} c-2z-A^T\pi \\ -b \end{pmatrix}.$$

Hence, $\Delta\pi$ is independent of x. We have from (6.3.3) that

$$\mu Z^{-2}\Delta z = -x-\Delta x+\mu Z^{-1}e = -\hat{x}+\mu Z^{-1}e.$$

Hence, Δz is also independent of x and there is no need to take the same step along Δz and $\Delta\pi$ as Δx. We are therefore in an analogous situation to the primal case, and we define the iterative sequence as

$$\begin{aligned} x_{k+1} &= x_k + \alpha\Delta x \\ z_{k+1} &= z_k + \alpha_z\Delta z \\ \pi_{k+1} &= \pi_k + \alpha_z\Delta\pi. \end{aligned}$$

Again, we may recur an additional variable $\bar{x}_k$ according to

$$\begin{aligned} \bar{x}_0 &= x_0 \\ \bar{x}_{k+1} &= \bar{x}_k + \alpha_z\Delta\bar{x}_k, \end{aligned}$$

with

$$\Delta\bar{x}_k = x_k + \Delta x_k - \bar{x}_k.$$

Exactly as in the primal case, if $\alpha_z = 1$ at iteration K, then $(\bar{x}_k, z_k, \pi_k)$ are dual feasible for $k > K$. Similarly if $\alpha = 1$ at iteration $\bar{K}$, then x_k is primal feasible for $k > \bar{K}$.

A similar convergence analysis to the primal case shows that

$$\begin{aligned} \lim_{k\to\infty} x_k &= \lim_{k\to\infty}\bar{x}_k = x^*(\mu) \\ \lim_{k\to\infty} z_k &= z^*(\mu) \\ \lim_{k\to\infty} \pi_k &= \pi^*(\mu). \end{aligned}$$

6.4 A Primal-Dual Method

If we premultiply the first n elements of f_p in (6.2.3) by X we obtain

$$Zx - \mu e = 0.$$

Consequently, the solution of (6.3.2) such that $x > 0$ is also the solution of the following system of equations:

$$f_{pd}(z, x, \pi) \equiv \begin{pmatrix} Zx - \mu e \\ c + Hx - z - A^T\pi \\ Ax - b \end{pmatrix} = 0. \tag{6.4.1}$$

These are termed primal-dual equations not because they can be shown to arise from a primal-dual form of LP, but simply because it is now desirable that both x and z remain strictly positive. The Jacobian of f_{pd} is given by

$$J_{pd} = \begin{pmatrix} X & Z & 0 \\ -I & H & A^T \\ 0 & A & 0 \end{pmatrix}.$$

If we apply Newton's method to (6.4.1), the following system is solved at each iteration:

$$J_{pd} \begin{pmatrix} \Delta z \\ \Delta x \\ -\Delta\pi \end{pmatrix} = - \begin{pmatrix} Zx - \mu e \\ c + Hx - z - A^T\pi \\ Ax - b \end{pmatrix}. \tag{6.4.2}$$

The corresponding KKT system is

$$\begin{pmatrix} H + X^{-1}Z & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta\pi \end{pmatrix} = - \begin{pmatrix} c + Hx - \mu X^{-1}e - A^T\pi \\ Ax - b \end{pmatrix}. \tag{6.4.3}$$

As in the primal and dual cases, we may obtain equations in just $\Delta\pi$:

$$A(H + X^{-1}Z)^{-1}A^T\Delta\pi = b - Ax + A(H + X^{-1}Z)^{-1}(g_B - A^T\pi). \tag{6.4.4}$$

A proof of convergence is no longer obvious since it is no longer transparent that restricting $x_k > 0$ and $z_k > 0$ will not inhibit convergence. If we introduce a separate step for z and π as before, we can show that the resulting sequence converges to the desired values provided the step for x is chosen as in the primal method. To appreciate why, observe that the right-hand side of (6.4.3) and (6.2.5) are identical.

Assume for the moment that x_k is feasible; then the right-hand side of both systems is $(A^T\pi - g_B, 0)$. Let V denote a basis for the null space of the columns of A, i.e., $AV = 0$. Since $A\Delta x_k = 0$, it follows that

$$\Delta x_k = V\Delta x_v,$$

where

$$V^T(H + X^{-1}Z)V\Delta x_v = -V^T g_B.$$

Premultiplying this equation by Δx_v^T gives

$$\Delta x_k^T g_B = \Delta x_v^T V^T g_B = -\Delta x_v^T V^T (H + X^{-1}Z)V\Delta x_v.$$

Since the largest eigenvalue of $V^T(H+X^{-1}Z)V$ is bounded and the smallest eigenvalue is bounded away from zero for all k, it follows that Δx is a direction of sufficient descent of $B(x,\mu)$. Given that we choose α to satisfy the Goldstein-Armijo condition, it follows from standard convergence analysis that $\lim_{k\to\infty} \|g_B\| = 0$. The convergence of $\{z_k\}$ and $\{\pi_k\}$ follows from Lemma 6.2.2.

We have assumed that x_k is feasible. A similar analysis based on the use of a merit function such as $M(x) = B(x,\mu) + \rho\|Ax - b\|_1$ makes this assumption unnecessary.

6.5 A Second Primal-Dual Method

If we premultiply the first n equations of (6.2.3) by $\frac{1}{\mu}Z^{-1}$, we get

$$X^{-1}Z^{-1}e - \frac{1}{\mu}e = 0.$$

Consequently, the solution of (6.2.3) such that $x > 0$ is also the solution of the following sytem of equations:

$$\bar{f}_{pd}(z,x,\pi) \equiv \begin{pmatrix} X^{-1}Z^{-1}e - \frac{1}{\mu}e \\ c + Hx - z - A^T\pi \\ Ax - b \end{pmatrix} = 0. \tag{6.5.1}$$

The Jacobian of $\bar{f}_{pd}$ is given by

$$\bar{J}_{pd} = \begin{pmatrix} -Z^{-2}X^{-1} & -X^{-2}Z^{-1} & 0 \\ -I & H & A^T \\ 0 & A & 0 \end{pmatrix}.$$

If we apply Newton's method to (6.5.1), the following system is solved at each iteration:

$$\bar{J}_{pd}\begin{pmatrix}\Delta z\\ \Delta x\\ -\Delta\pi\end{pmatrix} = -\begin{pmatrix}X^{-1}Z^{-1}e - \frac{1}{\mu}e\\ c + Hx - z - A^T\pi\\ Ax - b\end{pmatrix}. \tag{6.5.2}$$

The corresponding KKT system is

$$\begin{pmatrix}H + X^{-1}Z & A^T\\ A & 0\end{pmatrix}\begin{pmatrix}\Delta x\\ -\Delta\pi\end{pmatrix} = -\begin{pmatrix}c + Hx - 2z + \frac{1}{\mu}Z^2x - A^T\pi\\ Ax - b\end{pmatrix}. \tag{6.5.3}$$

Note the matrix on the right-hand side of 6.5.3 is identical to that of 6.4.4. Again, using the Schur complement of $H + X^{-1}Z$, we obtain equations in $\Delta\pi$:

$$A(H+X^{-1}Z)^{-1}A^T\Delta\pi = b - Ax + A(H+X^{-1}Z)^{-1}(c + Hx - 2z + \frac{1}{\mu}Z^2x - A^T\pi). \tag{6.5.4}$$

In this primal-dual method a proof of convergence is quite straightforward. We now have $\|\bar{f}_{pd}\| = \infty$ if any element of x or z is zero. Consequently, taking a step that satisfies the Goldstein-Armijo conditions ensures that the smallest singular value of $\bar{J}_{pd}$ is bounded away from zero and that the largest singular value is bounded. Unlike the three previous methods, it is not apparent that the step taken in the x variables may be different from that taken in z and π.

6.6 Affine Variants

In the LP case it can be shown that the vector

$$p = \lim_{\mu\to 0}\frac{\Delta x(\mu)}{\|\Delta x(\mu)\|}$$

exists. Algorithms based on using p as the search direction have been advocated by a number of researchers. A proof that such an algorithm converges is given by Vanderbei and Lagarias [VL90]. It is of some interest therefore to consider whether similar algorithms exist for the quadratic case. In this section we assume H is positive definite.

In the case of the primal method, simple inspection of the equations (6.2.5) shows that if H is nonsingular,

$$\begin{pmatrix}H & A^T\\ A & 0\end{pmatrix}\begin{pmatrix}p\\ -\Delta\pi\end{pmatrix} = -\beta\begin{pmatrix}c + Hx - A^T\pi\\ Ax - b\end{pmatrix}, \tag{6.6.1}$$

where β is a positive constant chosen to ensure $\|p\| = 1$. The search direction in the x variables always points towards the constrained stationary point of the QP problem with no bounds. Obviously such an algorithm will not in general converge to the desired solution.

In the case of the dual method we get

$$\begin{pmatrix} Z^2 & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} p \\ -\Delta\pi \end{pmatrix} = -\beta \begin{pmatrix} Z^2 x \\ Ax - b \end{pmatrix}. \tag{6.6.2}$$

Clearly any search direction that does not involve the Hessian of the quadratic is not likely to be sensible. There is somewhat more hope that the primal-dual methods will lead to sensible search directions since μ only occurs on the right-hand side of the Newton equations. For the first primal-dual method the corresponding KKT system is

$$\begin{pmatrix} H + X^{-1}Z & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} p \\ -\Delta\pi \end{pmatrix} = -\beta \begin{pmatrix} c + Hx - A^T\pi \\ Ax - b \end{pmatrix}, \tag{6.6.3}$$

which reduces to the usual primal-dual affine algorithm for LP when $H = 0$. Even though the right-hand side $c + Hx - A^T\pi$ does not tend to zero, such a method converges, though more slowly than when μ is smoothly reduced.

The second primal-dual method reduces to

$$\begin{pmatrix} H + X^{-1}Z & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} p \\ -\Delta\pi \end{pmatrix} = -\beta \begin{pmatrix} -Z^2 x \\ Ax - b \end{pmatrix}, \tag{6.6.4}$$

and in this case the right-hand side does tend to zero. It is not immediately apparent that this gives a sensible direction. However, there is no prima facie case to dismiss it.

6.7 The Case When $H + D$ is Trivially Invertible

If $H + D$ is trivially invertible (where D is a diagonal matrix), we may obtain Δx by first obtaining $\Delta\pi$ from equations (6.2.6), (6.3.5), (6.4.4) or (6.5.4). Such an approach is popular in the LP case, in spite of the fact that the Schur complements $A\bar{D}A^T$ (for various diagonal $\bar{D}$) are much more ill-conditioned than the KKT systems themselves. The approach is particularly attractive if $m \ll n$.

A number of important QP problems have an H that is of the required structure or may be converted to such a problem. Examples of two such objective functions are:

$$\min_{x \in \mathbb{R}^n} x^T x$$

and

$$\min_{x \in \Re^n} \quad \tfrac{1}{2} x^T M x + \tfrac{1}{2} x^T V V^T x,$$

where M is a positive definite diagonal matrix and V is an $n \times t$ matrix with $t << n$. In the second case it may appear that $H = M + VV^T$ is not trivially invertible. However if we introduce the variable $y = V^T x$, the problem is transformed into one whose Hessian is diagonal, at the cost of adding t variables and t general constraints to the problem.

Free Variables.

We may like to use this approach when H is only positive semidefinite. Under such circumstances $H + D$ will normally be positive definite. However, if there are some free variables it may be that $H + D$ is singular. We shall consider just the primal algorithm but the approach we advocate may be used in all the methods. Suppose that just x_j is a free variable. In place of $(f_p)_j = z_j - \mu/x_j$ we now have $(f_p)_j = z_j$. Provided the KKT system is solved, the effect this equation has on the Jacobian is of no consequence. However, in the KKT system in place of $H + \frac{1}{\mu} X^{-2}$ we now have $H + D$, where $d_j = 0$ and $d_i = \mu x_i^{-2}$ for $i \neq j$. Since $H + D$ may be singular, its Schur complement may not exist. A means of circumventing this difficulty is to replace the equation $z_j = 0$ by

$$e^{x_j} z_j = \mu.$$

We then have $d_j = z_j$. Since $z_j^*(\mu) = \mu e^{-x_j}$, at the solution we may keep $z_j > \epsilon > 0$. It follows that $H + D$ is nonsingular and its Schur complement exists.

6.8 No General Constraints

If there are no general constraints, the barrier subproblem is an unconstrained minimization problem. A problem of considerable interest in this category is bound-constrained linear least squares. For the primal method the subproblem is

$$\underset{x \in \Re^n}{\text{minimize}}\ B(x, \mu) \equiv c^T x + \tfrac{1}{2} x^T H x - \mu \sum_j \ln x_j. \tag{6.8.1}$$

Even if H is indefinite, such a problem may be solved by a modified Newton method based on a Cholesky factorization. For example, see Forsgren, Gill and Murray [FGM89a].

When H is positive definite as in the least-squares case, the equivalent reduction of equations (6.3.2), (6.4.1) and (6.5.1) is also possible. Note that when there are only simple bounds, degeneracy cannot arise in the primal or the dual.

6.9 Combination Methods

Any linear combination of the four sets of equations (6.2.3), (6.3.2), (6.4.1) and (6.5.1) may also be used to define a method. If the linear combination is such that the resulting function is infinite if any element of z or x is zero, then the convergence of Newton's method follows from an identical argument to that given for the second primal-dual method. To illustrate the approach, consider the following linear combination of the f_{pd} and $\bar{f}_{pd}$:

$$\mu \bar{f}_{pd} + \frac{1}{\mu} f_{pd} = 0.$$

The resulting Jacobian is

$$\begin{pmatrix} -\mu Z^{-2}X^{-1} + \frac{1}{\mu}X & \frac{1}{\mu}Z - \mu X^{-2}Z^{-1} & 0 \\ -I & H & A^T \\ 0 & A & 0 \end{pmatrix}.$$

If $v = \mu Z(Z^2X^2 - \mu^2 I)^{-1}(\mu e - 2Zx + \frac{1}{\mu}Z^2X^2e)$, the corresponding KKT system reduces to

$$\begin{pmatrix} H + X^{-1}Z & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta \pi \end{pmatrix} = - \begin{pmatrix} c + Hx - A^T\pi + v \\ Ax - b \end{pmatrix},$$

which has the same KKT matrix as the two individual primal-dual methods.

Chapter 7

QPS Format

7.1 Introduction

For linear programming problems, the MPS format [IBM76] has become the industry standard and it is recognized by all commercial mathematical programming systems. Unfortunately, there is no equivalent standard format for quadratic programming. In this chapter we define a format suitable for large, sparse quadratic programs of the form

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & c^T x + \frac{1}{2} x^T H x \\ \text{subject to} & b_1 \leq Ax \leq b_2 \\ & \ell \leq x \leq u. \end{array} \tag{7.1.1}$$

The basis of designing the format has been to adhere as closely as possible to the desirable features present in the MPS format.

7.2 The QPS File

A QPS file is required to define the quadratic program. The file contains names for the variables and constraints, as well as the nonzero elements of A, H, b_1, b_2, c, ℓ and u. As with an MPS file, a *fixed format* must be used, i.e., each item of data must appear in specific columns. The QPS file is divided into the following sections:

```
NAME
VARIABLES
⋮
HESSIAN
⋮
CVECTOR
⋮
ROWS
⋮
COLUMNS
⋮
ENDATA
```

Each section starts with a *header line* that is the section name beginning in column 1. The remaining data within a section has the following fixed format:

Columns	2.3	5...12	15...22	25...36	40...47	50...61
Contents	*Key*	*Name0*	*Name1*	*Value1*	*Name2*	*Value2*

In addition, "comment" lines are allowed; these have an asterisk '*' in column 1 and any characters in columns 2–22.

The NAME Section.

```
1..4  5..12  15...22
NAME          MYQPNAME  (for example)
```

This section constitutes just one line with the word NAME in columns 1–4, and a name for the problem in columns 15–22. This NAME line is normally the first one in the MPS file, but it may be preceded or followed by comment lines. The name for the problem is used to label the solu'ion output and it may be from 1 to 8 characters of any kind, or it may be blank.

The VARIABLES Section.

```
Key   Name0   Name1   Value1    Name2    Value2
2.3   5...12  15...22 25...36   40...47  50...61   (fields)

VARIABLES
 LO   VAR01           1.0
 UP   VAR02                              3.5
 RA   VAR03           -4.6               90.7      (for example)
 MI   VAR04           200.0
 FR   VAR05
```

This section contains one line for each variable. Field *Name0* gives the variable an 8-character name, and field *Key* defines the variable type by specifying its bounds. The default bounds on all variables x_j (excluding slacks) are $0 \leq x_j \leq \infty$. If necessary, the default values 0 and ∞ can be changed to $l \leq x_j \leq u$. The various bound-types are:

Key	*Bound-type*
FR	Free variable
FX	Fixed variable
MI	Minus infinity
PL	Plus infinity
RA	Range
LO	Lower bound
UP	Upper bound

All these bound specifications overwrite the default values. The numerical values for the bounds are specified in the fields *Value1* and *Value2*. For instance, they define respectively the lower and upper bound for the range type. Incidentally, only this type (RA) needs two values. The remaining bound-types need at most one. The numerical value, say β, can appear either in field *Value1* or *Value2*, but not in both. That is, one of them should be blank in the input file; otherwise the sum of the two values will be taken as the bound. In all sections of the QPS file, fields *Value1* and *Value2* are read using Fortran format E12.0. This allows values to be entered in several forms; for example, 1.2345678, 1.2345678E+0, and 12.345678E-1 all represent the same number. It is usually best to include an explicit decimal point. In some computer systems, spaces within the value field are taken as 0's. Hence, if an exponent like E-3 is used, it must be *right-justified* in the value field.

The types LO and UP are modifiers of the default values. The type LO will overwrite *only* the default lower bound. Analogously, the UP type will overwrite *only* the default upper bound. Every variable must appear in the VARIABLES section and at most once. If a variable is missing, it will not be recognized when its name appears in later sections. If there are duplicates (more than one entry with the same variable name) an error message will be issued indicating so. For instance, the following combination cannot be used to define a bounded variable:

```
LO  VAR123  -5.0
UP  VAR123   7.0
```

Instead, the range bound-type should be used:

```
RA  VAR123  -5.0  7.0
```

For the bound-types other than RA, let l and u be the default lower and upper bounds respectively, and let x and β be the variable and bound specified in the VARIABLES section. The various types allowed result in the following bounds:

Key	*Bound-type*	*Resulting bounds*
LO	Lower bound	$\beta \leq x \leq u$
UP	Upper bound	$\ell \leq x \leq \beta$
FX	Fixed variable	$\beta \leq x \leq \beta$
FR	Free variable	$-\infty \leq x \leq \infty$
MI	Minus infinity	$-\infty \leq x \leq \beta$
PL	Plus infinity	$\beta \leq x \leq \infty$

Applying these rules to the examples at the beginning of this section gives the following bounds:

$$
\begin{array}{rcrcr}
1.0 & \leq & \texttt{VAR01} & \leq & \infty \\
-\infty & \leq & \texttt{VAR02} & \leq & 3.5 \\
-4.6 & \leq & \texttt{VAR03} & \leq & 90.7 \\
-\infty & \leq & \texttt{VAR04} & \leq & 200.0 \\
-\infty & \leq & \texttt{VAR05} & \leq & \infty
\end{array}
$$

Note that types FR, MI, or PL should always be used to specify "infinite" bounds: they imply values of $\pm 10^{20}$, which are treated specially at certain times. Note also that types MI and PL are different from the MPS format.

The HESSIAN Section.

```
    Name0   Name1    Value1   Name2    Value2
    5...12  15...22  25...36  40...47  50...61  (fields)

HESSIAN
    VAR01   VAR01      1.0    VAR03     -3.0
    VAR01   VAR11      2.0    VAR21      4.5
    VAR01                     VAR30      5.5
    VAR11   VAR12     -1.0    VAR15      80.
    VAR11   VAR23     -8.4
    VAR25                     VAR25      3.3
    VAR25   VAR33      2.2
    VAR22   VAR25      3.3    VAR22      0.4
```

This section contains the nonzero elements of the Hessian of the quadratic program. To specify the indices for the nonzero element h_{ij}, the names associated with the variables x_i and x_j in the previous section are used. Each line defines at most two elements, h_{ij} and h_{kj} say. Fields *Name0*, *Name1* and *Name2* contain the names for the jth variable, ith constraint and kth constraint respectively. Fields *Value1* and *Value2* contain the numerical values h_{ij} and h_{kj} respectively.

Since the Hessian is symmetric, only the elements in either the lower or the upper triangular block need to be specified. If both elements h_{ij} and h_{ji} are entered, an error message will be issued indicating that there is a duplicate.

Notice that the element a_{ij} of the constraint matrix involves indices of two distinct sets: the VARIABLE names and the ROW names. In contrast, the indices of the element h_{ij} of the Hessian are both VARIABLE names. Therefore, the ordering of the variables becomes critical when entering the Hessian. It is more efficient and computationally convenient to have an ordering already assigned to the variables before the Hessian is read in. For this reason, we chose the VARIABLE section to be the one that defines the ordering of the variables, namely the order in which they appear.

The nonzero elements of the Hessian must be entered by column; that is, they must appear grouped together before the elements of the next column in the QPS file. If a column has several nonzeros it does not matter what order they appear in. Columns may also be specified in any order. However, sorting the Hessian elements can be avoided if the columns of the Hessian are entered in the order defined in the VARIABLES section, and only the elements in the lower triangular block are specified. The implementation recognizes such a case and economizes accordingly.

The CVECTOR Section.

```
                Name1     Value1    Name2     Value2
                15...22   25...36   40...47   50...61   (fields)
CVECTOR
                VAR01       1.0     VAR02      -3.0
                                    VAR50       2.2
                VAR33      -1.3     VAR20      50.1
                VAR40       2.2
```

This section contains the nonzero elements of the vector c involved in the linear term of the quadratic objective function.

For each nonzero component of c, say c_j, field *Name1* (or *Name2*) contains the name for the jth variable and field *Value1* (or *Value2*) contains the numerical value c_j. At most two nonzero components of c can be specified per line.

The ROWS Section.

```
Key   Name0   Name1   Value1   Name2   Value2
2.3   5..15   15..22  25..36   40..47  50..61   (fields)
ROWS
  E   ROW01             -3.0
  G   ROW02              2.2
 RA   ROW03             -1.3             50.1
  L   ROW04                              100.2
```

In this section we have kept the terminology used in linear programming, where the general constraints are referred as *rows*. The ROWS section contains one line for each constraint.

Key	*Row-type*	*Resulting constraint*
E	=	$a^T x = \beta$
G	$\geq$	$a^T x \geq \beta$
L	$\leq$	$a^T x \leq \beta$
RA	range	$\ell \leq a^T x \leq u$

The 1-character Key may be in column 2 or 3. Row-types have the natural interpretation in terms of a linear function $a^T x$ and bounds β, l, and u. Nonzero elements of the row-vector a will appear in appropriate parts of the next section.

The COLUMNS Section.

```
    Name0  Name1   Value1   Name2   Value2
     5..12 15..22 25...36  40..47  50...61  (fields)
COLUMNS
    VAR01  ROW01     1.0    ROW02    -3.0
    VAR01  ROW05     2.5    ROW03    1.123
    VAR02  ROW02  -11.123
    VAR02                   ROW20    7.777
    VAR03  ROW01  1.2E-02
```

This section defines the constraint matrix. It uses the name assigned to each variable x_j in the VARIABLES section and lists the nonzero entries a_{ij} in the corresponding column of the constraint matrix, using the same fields as the HESSIAN section. As in the latter, the nonzero elements within a particular column must be grouped together. If a column has several nonzeros, it does not matter what order they appear in (as long as they all appear before the next column). Columns in the constraint matrix correspond to variables in the problem; hence the VARIABLES section also defines an ordering for them. As in the HESSIAN section, it should be possible to enter columns in any order, and a sort would again be required. This option has not yet been implemented.

In general, *Name0* is the column/variable name, and *Name1*, *Value1* give a row name and a value for some element in that column. If there is another nonzero element in that column, it may appear as *Name2*, *Value2* on the same line, or may be on the next line. If either *Name1* or *Name2* is blank, the corresponding value is ignored.

There is no need to specify columns for the slack variables; they are incorporated implicitly.

If the columns appear in the same order as in the VARIABLES section, the implementation recognizes that the constraint data need not be sorted.

7.3 Differences from MPS Format

The design of the QPS format combines the desirable features present in the MPS format for linear programs, with the necessary extensions to be able to specify a quadratic program. In this section we describe the differences between the two formats.

In the MPS format the names for the variables are specified in the COLUMNS section and their bounds are defined in the BOUNDS section. In the QPS format

there is no BOUNDS section. Instead, a new section, the VARIABLES section, has been introduced where both names and bounds are specified.

With the MPS format it is possible to define several related problems in just one file, by specifying several right-hand sides (for example) or several bound sets. At the moment only one problem can be specified in a QPS file.

In order to minimize the storage required, the QPS format does not allow a VARIABLE name to be the same as a ROW name.

The general constraints are referred to as *rows* in both formats. In the MPS format a constraint is defined as follows: its name and type are specified in the ROWS section, its right-hand side (if nonzero) is defined in the RHS section and for the case where the constraint has lower and upper bounds, they are specified implicitly in the RANGES section. In the QPS format the ROWS section contains not only the name and type of a constraint, but also its bounds. Hence, the RHS and RANGES sections are no longer needed.

Unlike the MPS format, in the QPS format the COLUMNS section does not include the vector c involved in the linear term $c^T x$ of the objective function. Instead, it is defined in the CVECTOR section.

A new section has been introduced in order to specify the Hessian matrix. The format used to express this matrix is identical to the one used in the COLUMNS section of the MPS format to define the constraint matrix.

In the MPS format there are only two sections where the variables appear: the COLUMNS section and the BOUNDS section. The ordering assigned to the variables corresponds to that in which they appear in the COLUMNS section. In the QPS format, there are three sections involving the variables: the VARIABLES section, the HESSIAN section, and the COLUMNS section. The VARIABLES section must come first and it is used to define the ordering of the variables. This extension proved necessary because the HESSIAN and COLUMNS sections may not contain a complete list of variables. (In a general QP, both H and A may contain empty columns.)

If the HESSIAN section was required to contain a complete list of variables, it would seem that we could use it as the one that defines the ordering in the variables. Unfortunately it would no longer be possible to avoid sorting the Hessian elements if we insist on checking for duplicates.

7.4 Test Problems

Linear programs are a subset of the problems under study. Hence, a code has been developed that takes as input the specification of a linear programming problem in MPS format and produces as output the corresponding definition in QPS format. In this way, test problems can be generated from the set of linear programs available in the software library *netlib* [Gay85].

Finally we point out that the QPS format is particularly appropriate to define problems that result from adding a quadratic term to an existing linear program, whose specification in the MPS format is already available. For instance, regularized linear programs can be defined in a straightforward manner, i.e., those with objective function of the form

$$c^T x + \tfrac{1}{2}\rho\|x - x_0\|^2,$$

where $\rho \geq 0$. (By varying the size of ρ a series of test cases can be generated with varying degrees of nonlinearity.) On the other hand, a different format might be more suitable when the sparsity pattern of the constraints and the Hessian of the Lagrangian function are related. Quadratic programs of this form arise in the optimization of electrical power transmission (the Optimal Power Flow problem [SAM80]) when one applies a sequential quadratic programming method using exact second derivatives.

In general, the QPS format will be appropriate as long as the Hessian can be expressed using one of the common formats for sparse matrices, namely, a list of triples (i, j, h_{ij}).

7.5 Example

```
* This is an example of a QP in QPS format.
*.......1.........2.........3.........4.........5.........6
NAME          DIET
VARIABLES
 RA OATMEAL                0.0                      4.0
 RA CHICKEN                0.0                      3.0
 RA EGGS                   0.0                      2.0
 RA MILK                   0.0                      8.0
 RA PIE                    0.0                      2.0
 RA PORKBEAN               0.0                      2.0
 UP BREAD                  2.0
 LO BUTTER                 9.5
 FX PASTA                                          33.3
 MI COFFEE                 6.57
 PL TUNA                                           4.3
HESSIAN
    OATMEAL   OATMEAL      2.0        EGGS          5.0
    OATMEAL   PIE          0.5
    EGGS      PORKBEAN     7.0        EGGS          4.0
    PORKBEAN  PORKBEAN     0.7        PASTA        30.
    PORKBEAN  COFFEE      10.         TUNA         50.
    BREAD     BUTTER      60.         PASTA        70.
    PASTA     PASTA       80.         COFFEE       90.
    PASTA     TUNA       100.
CVECTOR
    COST      OATMEAL      3.        CHICKEN       24.
    COST      EGGS        13.        MILK           9.
    COST      PIE         20.        PORKBEAN      19.
ROWS
 RA ENERGY                2000.                   12000.
 G  PROTEIN                 55.
 G  CALCIUM                800.
COLUMNS
    OATMEAL   ENERGY      110.        PROTEIN        4.
    OATMEAL   CALCIUM       2.
    CHICKEN   ENERGY      205.        PROTEIN       32.
    CHICKEN   CALCIUM      12.
    EGGS      ENERGY      160.        PROTEIN       13.
    EGGS      CALCIUM      54.
    MILK      ENERGY      420.        PROTEIN        8.
    MILK      CALCIUM     285.
    PIE       ENERGY      420.        PROTEIN        4.
    PIE       CALCIUM      22.
    PORKBEAN  ENERGY      260.        PROTEIN       14.
    PORKBEAN  CALCIUM      80.
ENDATA
```

Chapter 8

Implementation Issues and Numerical Results

8.1 Introduction

An implementation of the barrier algorithm BARQP was developed for the solution of both linear and quadratic programming problems in the standard form

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{minimize}} \; Q(x) = c^T x + \tfrac{1}{2} x^T H x \\ &\text{subject to } Ax = b, \quad l \leq x \leq u. \end{aligned}$$

The implementation treats A and H as sparse matrices and b, c, l and u as sparse vectors, specified in the QPS format of Chapter 7. The algorithm follows closely that described in Chapters 3 and 4.

Barrier algorithms are sensitive to many of the parameters used in their definition. The development of a practical implementation involves considerable experimentation to determine values for these critical parameters that strike a balance between efficiency and reliability. In this chapter we describe the main parameters, such as the initial μ, the final μ, and the distance of the initial point from its bounds.

Not all the features are currently incorporated in the code. For example, we have yet to include procedures for computing directions of negative curvature. As previously mentioned we hope to reduce the need for such directions by a suitable choice of parameters. However, any complete implementation would need such procedures as a safeguard.

One of the aims has been to explore the performance of the barrier algorithm on a set of real-world problems, and to compare its execution time with that of MINOS, a well known code for sparse linear and nonlinear programming [MS83]. These results are not meant as an accurate reflection of the performance of a barrier QP algorithm but are intended to serve as an indication of the algorithm's potential.

8.2 Free Variables

The form of the problem discussed so far assumes that a variable has one or two bounds. In practice some variables may have no bounds. Such variables are termed *free* and they are not included in the barrier term. The resulting zero diagonals present

a difficulty if the KKT system is solved via the Schur complement (see Equation (6.2.6)). (In contrast, free variables are an asset to the simplex algorithm since they may be included in the basis throughout.) If the KKT system is solved directly, the only difficulty is that the pivot order chosen in the symbolic factorization may be revised in the numerical phase. Interestingly for QP's, free variables may not present a difficulty even if the KKT system is solved by the Schur complement approach, since $H_B = H + \mu D$ could be nonsingular even if D is singular.

8.3 Feasibility

Interior-point methods differ from the simplex method in satisfying the bounds $\ell < x < u$ throughout and declaring "primal feasibility" when $Ax = b$. (The primal simplex method satisfies $Ax = b$ throughout and achieves feasibility when $\ell \leq x \leq u$.)

Early interior-point implementations for LP achieved primal feasibility by setting up a Phase 1 problem of the form

$$\begin{aligned} \min_{x,\xi} \quad & \xi \\ \text{subject to} \quad & Ax + \xi r_0 = b, \\ & \ell \leq x \leq u, \quad \xi \geq 0. \end{aligned}$$

(e.g. [Kar84,GMS*86,Mar89]), where $r_0 = b - Ax_0$ is the residual for some given initial point x_0 with $\ell < x_0 < u$. A refinement is to use a "composite objective" of the form $\omega c^T x + \xi$ for some $\omega \geq 0$, in order to drive the feasibility phase closer to optimality for the original problem.

This approach was followed in the initial versions of our barrier QP method. The Phase 1 problem took the form

$$\begin{aligned} \min_{x,\xi} \quad & \omega(c^T x + \tfrac{1}{2} x^T H x) + \xi \\ \text{subject to} \quad & Ax + \xi r_0 = b, \\ & \ell \leq x \leq u, \quad \xi \geq -1, \end{aligned}$$

and feasibility was declared when ξ decreased from $\xi_0 = 1$ to zero. Unfortunately r_0 is typically a dense vector, and the parameter ω must be chosen carefully. Also, provision must be made to reduce ω at any iteration if ξ does not decrease.

These difficulties are avoided by treating the barrier method as the application of Newton's method to the QP feasibility and optimality conditions, as described in Chapter 4. The result is a "single-phase" primal method in which the merit function

$$M(x, \mu) = \|g_L\| + \|r\|$$

is reduced at each iteration. In general, $r = b - Ax$ is reduced to zero *before* g_L, and remains zero thereafter (since the equations $Ax = b$ are linear). Thus, feasibility may be achieved even if the algorithm is terminated prematurely.

8.4 Solution of the KKT Systems

Our main software tool for solving the KKT systems is the Harwell Subroutine Library package MA27, due to Duff and Reid [DR82,DR83]. This is a multifrontal code for solving sparse symmetric linear equations $Ky = z$. When K is sufficiently positive definite, MA27 computes a sparse Cholesky factorization $K = U^TDU$, where D is diagonal and U is unit upper triangular. Otherwise, MA27 uses an implementation of the Bunch-Kaufman algorithm [BK77] to compute $K = U^TDU$, where D is now block-diagonal with blocks of dimension 1 or 2.

Two of the design aims for MA27 were to be competitive with existing Cholesky codes on definite systems, and to be *nearly* as efficient on indefinite systems. The first aim was achieved and the second was at least partially achieved, in the sense that the package has proved to be efficient on systems that are only slightly indefinite (i.e. systems where nearly all eigenvalues have the same sign). Unfortunately, the KKT systems arising in our context are usually very indefinite (unless $m \ll n$). The analyze phase of MA27 assumes that K is definite and in particular that all diagonals of K are nonzero. When this is not true, the tentative pivot order is sometimes drastically revised in the numerical phase, with a consequent increase in nonzeros in the numerical factors. The associated increase in computation time is substantial in the case of KKT systems.

In the near future, we expect a significant improvement will be realized using a new version of the package to be called MA47; see Duff *et al.* [DGR*89].

In general, Aasen's tridiagonalization method [Aas71] is considered competitive with the Bunch-Kaufman approach for solving dense indefinite systems. Aasen's method computes a factorization of the form $K = U^TTU$ where T is tridiagonal. However, we do not know of a sparse implementation.

8.5 Rank Deficiency in the Constraint Matrix

In many practical applications, the constraint matrix is either poorly conditioned or singular. The associated KKT system is then also ill-conditioned or singular. (Consider the case when the rows of the constraint matrix A are linearly dependent.)

When A is ill-conditioned, we consider the following modified KKT system:

$$\begin{pmatrix} H_B & A^T \\ A & -\delta I \end{pmatrix} \begin{pmatrix} -p \\ q \end{pmatrix} = \begin{pmatrix} g_L \\ r \end{pmatrix}, \tag{8.5.1}$$

where $\delta > 0$, $g_L = c^T + Hx_k - A^T\pi_k$ and $r = b - Ax_k$. In most cases the modified system will be better conditioned when A is poorly conditioned. If δ is small we would not expect the convergence of Newton's method to be significantly affected. In practice we have used δ values in the range $(10^{-8}, 10^{-4})$. Systems of the form (8.5.1) have been studied in the context of sequential quadratic programming by [Mur69,Big75,Gou86b].

The modified system corresponds to the following perturbed QP:

$$\begin{aligned} \min_{p,y} \quad & c^T p + \tfrac{1}{2} p^T H_B p + \tfrac{1}{2}\delta y^T y \\ \text{subject to} \quad & Ap - \delta y = r. \end{aligned} \tag{8.5.2}$$

We observe that the dual to (8.5.2) is

$$\begin{aligned} \min_{p,y} \quad & r^T y + \tfrac{1}{2} p^T H_B p + \tfrac{1}{2}\delta y^T y \\ \text{subject to} \quad & A^T y + H_B p = c. \end{aligned} \tag{8.5.3}$$

In this formulation we see that the modification term serves to bound the Lagrange multipliers, which are unbounded when A is rank deficient. (The modification can therefore be viewed as a trust-region treatment of the Lagrange multipliers.)

We conclude this discussion by simply stating that the search direction given by the modified system is usually still a descent direction for the merit function given in Chapter 3. More specifically,

$$(p^T \;\; q^T)\nabla M(x,\pi) = -\|g_L\| - \|r\| - \frac{\delta}{\|r\|} q^T r,$$

which is negative if $q^T r \geq 0$ or δ is sufficiently small.

8.6 Simple Extrapolation

Let x and $g_L(x,\mu)$ be the final point and gradient obtained for a particular barrier parameter μ. When μ is reduced, we generally retain x as the starting point for the new subproblem. Unfortunately, $\|g_L\|$ inevitably suffers a sharp increase. For example, consider a variable x_j that is close to its lower bound of zero, and suppose μ changes to $\beta\mu$ $(0 < \beta < 1)$. By definition, the j-th component of g_L changes by the amount

$$\mu/x_j - \beta\mu/x_j = (1-\beta)\mu/x_j \approx (1-\beta)z_j,$$

where z_j is the j-th element of $c + Hx - A^T\pi$. If $z_j \gg 0$ and $\|g_L\|$ was previously small, we see that $\|g_L\|$ will no longer be small.

A simple means for counteracting this effect is to change x_j to βx_j, i.e., to reduce "small" elements of x by the same factor as μ, so that the corresponding gradient element does not change. Our experience has been that this strategy can reduce the number of minor iterations needed to solve each of the subproblems, as long as a conservative test is used to judge whether x_j is "close" to a bound. (The main danger lies in moving the "wrong" elements of x towards their bounds.)

8.7 Conventional Extrapolation

Let $x(\mu_k)$ be the solution to the nonlinear subproblem $B(x, \mu_k)$. As described in Fiacco and McCormick [FM68], if μ_k were steadily reduced to zero, the infinite sequence of points $\{x(\mu_k)\}$ would define a trajectory leading to x^*, the solution of the original problem. In practice we estimate $x(\mu_k)$ for only a few points on the trajectory. Using conventional extrapolation we can define an approximation to $x^* \equiv x(0)$ by extrapolating the solutions of two subproblems. Consider the expansion

$$x(\mu_k) \qquad + \gamma\mu_k + O(\mu_k^2), \tag{8.7.1}$$

where $\gamma = x'(0)$, and let $x(\mu_1)$ and $x($ $)$ be the solution of subproblems $B(x, \mu_1)$ and $B(x, \mu_2)$ respectively, with $\mu_1 > \mu_2$. From (8.7.1) we obtain

$$\mu_1 x(\mu_2) - \mu_2 x(\mu_1) = (\mu_1 - \mu_2)x^* + O(\mu_1^2),$$

so that

$$x^* \approx \frac{1}{(\mu_1 - \mu_2)}(\mu_1 x(\mu_2) - \mu_2 x(\mu_1)).$$

Since $x^* = x(\mu_2) + p$ for some p, we can define an approximate step to the solution as follows:

$$\begin{aligned} p &= x^* - x(\mu_2) \\ &\approx \frac{\mu_1 x(\mu_2) - \mu_2 x(\mu_1) - (\mu_1 - \mu_2)x(\mu_2)}{(\mu_2 - \mu_1)} \\ &= \frac{\mu_2}{(\mu_1 - \mu_2)}(x(\mu_1) - x(\mu_2)). \end{aligned}$$

When μ is decreased in a linear fashion, i.e. $\mu_2 = \beta\mu_1$ with $0 < \beta < 1$, we have

$$p \approx \frac{\beta}{(1 - \beta)}(x(\mu_2) - x(\mu_1)), \tag{8.7.2}$$

which can be used as a *search direction* to obtain an estimate $x^* \approx x(\mu_2) + \alpha p$ for some positive steplength α. Note that nothing is lost if the search proves unsuccessful, since the effort expended in the search is negligible.

In the early iterations when we are far from the solution, it may seem better to extrapolate to $x(\mu_3)$, the solution of the next subproblem $B(x, \mu_3)$, instead of to x^*.

Analogously to (8.7.1) we have

$$x(\mu_k) = x(\mu_3) + \gamma(\mu_k - \mu_3) + O(\mu_k - \mu_3)^2,$$

where $\gamma = x'(\mu_3)$. Hence,

$$(\mu_2 - \mu_3)x(\mu_1) - (\mu_1 - \mu_3)x(\mu_2) = (\mu_1 - \mu_2)x_3 + O(\mu_1^3),$$

so that

$$x(\mu_3) \approx -\frac{\beta_{23}}{\beta_{12}}x(\mu_1) + \frac{\beta_{13}}{\beta_{12}}x(\mu_2), \tag{8.7.3}$$

where $\beta_{ks} = \mu_k - \mu_s$. As before, when μ decreases linearly, (8.7.3) becomes

$$x(\mu_3) \approx -\frac{1}{\beta}x(\mu_1) + \frac{(\beta+1)}{\beta}x(\mu_2).$$

Since $x(\mu_3) = x(\mu_2) + \bar{p}$ for some $\bar{p}$, we can define an approximate step to $x(\mu_3)$ as

$$\begin{aligned} \bar{p} &= x(\mu_3) - x(\mu_2) \\ &= \frac{\beta_{32}}{\beta_{21}}(x(\mu_2) - x(\mu_1)). \end{aligned} \tag{8.7.4}$$

When μ is decreased in a linear fashion, (8.7.4) becomes

$$\bar{p} \approx \beta(x(\mu_2) - x(\mu_1)). \tag{8.7.5}$$

Comparing (8.7.2) and (8.7.5), we see that the extrapolation to x^* or to $x(\mu_3)$ leads to the same search direction (to within a scalar factor). Hence the only difference can be that the linesearch may give a non-equivalent steplength.

8.8 Parameter Selection

Within the barrier framework, a number of parameters must be specified. In general, we have found that the method is somewhat sensitive to this parameter selection, though there is a fairly broad range of suitable choices. In this section we summarize the criteria used in our barrier method and highlight some of the issues.

- **Initial barrier parameter μ_0**

  This parameter must be normalized by the number of logarithmic terms and the initial function value. That is, for a specified value $\hat{\mu}_0$ we take

$$\mu_0 = (\frac{\hat{\mu}_0}{n})F(x_0).$$

This corresponds to setting the weight of the logarithmic perturbation relative to the original QP function value $F(x_0)$. In general, if μ_0 is too large, a number of iterations will be wasted. That is, we solve a series of subproblem that are distant from the one we are interested in.[1] If, on the other hand, μ_0 is chosen too small, the iterates tend to progress near the boundary of the feasible region similarly to an active-set method. In practice, a range of values for $\hat{\mu}_0$ between 1 and 10^3 work reasonably well on most problems.

- **Final barrier parameter μ_f**

  As discussed in Chapter 3, the relative accuracy of the final answer is proportional to the size of the smallest μ that is used in minimizing $B(x,\mu)$. Thus, for a specified value $\hat{\mu}_f$,

  $$\mu_f = (\frac{\hat{\mu}_f}{n})F(x_0)$$

  yields $|\log(\hat{\mu}_f)|$ digits of accuracy. We have used $\hat{\mu}_f = 10^{-6}$.

- **Initial point x_0**

  The provision of a "good" initial point is critical to the performance of the algorithm. In general, both feasibility as well as optimality should be considered. Ideally, the initial point should be chosen near the solution of the minimization problem for the corresponding choice of μ. If, for example, approximate values of the variables are known, they can be used for an initial point. Unfortunately, this information is normally not known and we must be satisfied with a "rough" initial point that is judiciously far from the bounds.

  For the barrier method, it is necessary that the initial point be feasible with respect to the inequality constraints. Since the inequalities are only simple bounds, this is easily accomplished. However, feasibility of the equality constraints as well as the overall optimality conditions should be considered when determining the starting point.

  A number of possible strategies have been proposed in conjunction with the barrier method [Mar89]. One possibility is to choose x_0 such that

  $$\|x_0\| \approx \|\bar{b}\|,$$

  where $\bar{b}$ is constructed from positive lower bounds and negative upper bounds as follows:

  $$\bar{b} = b - \sum_{\ell_j>0} \ell_j a_j - \sum_{u_j<0} u_j a_j.$$

[1] It should be noted that even when the barrier function is far from the original objective function, significant progress can be made toward feasibility.

If the matrix A is well-scaled, this choice generates a starting point that is probably of the same magnitude as the true solution. Other possibilities include using some kind of greedy algorithm to choose each component of x_0 such that the initial residual $r_0 = b - Ax_0$ is to some extent minimized.

In our implementation, the initial point has been chosen to encourage rapid convergence to the solution of the first barrier subproblem. To illustrate some of the issues involved, consider the barrier function

$$B(x) = F(x) - \mu \sum_{j=1}^{n} \ln(x_j) \tag{8.8.1}$$

corresponding to the bounds $x \geq 0$. Let us first assume that x_0 is chosen to lie "close" to the bounds (i.e., $\|x_0\| \ll 1$). In a neighborhood of the bounds, the objective function is highly nonlinear; that is, the nonlinear logarithmic term dominates the overall function value. This nonlinear behavior can be seen from (4.7.6),

$$\bar{g}_L = (1 - \alpha)g_L + t(\alpha),$$

which expresses the new gradient as a function of the old gradient and the current point. The last term $t(\alpha)$ accounts for the nonlinearity of the function. We can see from (4.9.10) that when some of the x_j's are near their bounds, this nonlinear term will dominate the equation (unless the corresponding p_j's are small). Newton's method proceeds by forming a linear approximation to $t(\alpha)$. However, the linear approximation does not accurately capture the highly nonlinear behavior of the function when some components of x are near their bounds. Therefore, Newton's method cannot be expected to converge quickly (as is observed experimentally for many problems).

Now consider the converse situation; that is, assume that the starting point lies far from the bounds. In this case, x_0 is probably far from the true solution (assuming that some components of x^* lie on their bounds). However, the gradients of the logarithmic terms in (8.8.1) are now almost negligible. Unfortunately, it is these terms that force the iterates to pass through the center of the feasible region (as opposed to moving along the boundary). Now that they are small, the iterates tend to move toward the perimeter of the feasible region and thus require more iterations. Roughly speaking, a typical sequence of iterates might be as follows:

a) The initial point is far from all bounds.

b) The search direction is chosen as if the problem were unconstrained (since the barrier terms are negligible).

c) The steplength is chosen so that the next iterate is near one bound but far from all others.

d) The computation of the third point proceeds in a similar fashion to the computation of the second point. Since only one barrier term is significant, the search direction is computed almost as if only one bound were present.

Similarly for the subsequent iterates. To some extent the algorithm acts like an active-set method where the initial working set is empty.

To overcome these difficulties, we follow [Mar89] and introduce a linear term into the barrier function. Specifically, consider the QP

$$\begin{aligned} \min_x \quad & c^T x + \tfrac{1}{2} x^T H x \\ \text{subject to} \quad & Ax = b, \quad l \le x, \end{aligned}$$

and the logarithmic/linear transformation

$$\tilde{B}(x,\mu) = \min_x \; c^T x + \tfrac{1}{2} x^T H x - \mu(\sum_{j=1}^{n} \ln(x_j - l_j) - \nu \sum_{j=1}^{n} (x_j - l_j)),$$

where $\nu > 0$ (e.g., $\nu = 10^{-4}$). Notice that $\tilde{B}(x,\mu)$ again approaches the true solution as μ approaches zero. The advantage of this modified barrier transformation is that good initial points can be more easily found. In particular, the new function $\tilde{B}(x,\mu)$ has the property that

$$\lim_{\mu \to \infty} (x_j^*(\mu) - l_j) = \frac{1}{\nu}.$$

This is in contrast to the standard barrier function, which is unbounded. A second advantage is that when we are far from the bounds, the gradients of the linear terms (though small) are not negligible like those of the logarithmic terms. This has the advantage of making the code somewhat less sensitive to the point being far from the bounds. (See [Mar89] for some computational experiments on the sensitivity of the initial point for this code.) Therefore, we choose the initial point (the x_j's) by the following criteria:

$$x_j = \begin{cases} 0 & \text{if } x_j \text{ is a free variable,} \\ u_j - 1/\nu & \text{if } x_j \text{ has only an upper bound,} \\ l_j + 1/\nu & \text{if } x_j \text{ has only an lower bound,} \\ \frac{1}{2}(u_j + l_j) & \text{if } x_j \text{ contains upper and lower bounds.} \end{cases}$$

Experimentally we have found that this choice of x_0 behaves well over a wide range of problems.

- **Merit function**

  It can be shown that the search direction generated by the KKT system is a descent direction for the class of merit functions given by

  $$M(x,\pi) = f(x) - \pi^T r + \tfrac{1}{2}\omega\|r\|^2,$$

  where ω is a weighting parameter (see Chapter 3 for the case $\omega = 1$). In practice, we find that it is generally better to favor feasibility in deciding when to backtrack. Typically we choose $\omega = 10$.

- **Adjustment of μ**

  The strategy adopted for adjusting μ is similar to that described in [GMS*86]. Specifically, μ is decreased by a factor β if either $\|(g_L, r)\|$ is reduced by a factor γ or a maximum of ten iterations is reached. In the results reported, both β and γ were 0.3. The efficiency of the algorithm was not unduly sensitive to this choice. However, a more elaborate strategy may prove to be more efficient; for example, μ could be reduced by a factor that depends on the current point.

8.9 MINOS

MINOS is a Fortran-based mathematical programming system designed to solve a wide variety of large-scale linear and nonlinear programs [MS83]. For our comparisons we make use of the facilities for solving optimization problems with a smooth nonlinear objective and sparse linear constraints [MS78].

As in the implementation of our barrier algorithm, linear constraints are defined in the form $Ax = b, l \le x \le u$. A nonlinear objective $F(x)$ is represented by a Fortran subroutine that computes both $F(x)$ and its gradient $g(x)$ for any given feasible x.

For linearly constrained optimization, MINOS uses a reduced-gradient algorithm. This is an active-set method that computes a search direction using a null-space approach. (See Chapter 2 for a discussion of active-set and null-space methods.) The constraint matrix is partitioned as $A = \begin{pmatrix} B & S & N \end{pmatrix}$ where B is nonsingular. The associated variables are termed basic, superbasic and nonbasic respectively. The active-set matrix and the null-space matrix then take the form

$$\widehat{A} = \begin{pmatrix} B & S & N \\ & & I \end{pmatrix}, \qquad Z = \begin{pmatrix} B^{-1}S \\ I \\ 0 \end{pmatrix},$$

which satisfy $\widehat{A}Z = 0$ as required.

MINOS maintains a quasi-Newton approximation to the reduced Hessian in the form $Z^THZ \approx R^TR$, where R is upper triangular. The key equations for computing a search direction are therefore

$$R^TRp_S = -Z^Tg, \qquad p = Zp_S.$$

A sparse LU factorization of B allows Z^Tg and Zp_S to be computed efficiently, but R is stored as a dense triangular matrix of order s (the number of superbasic variables). Quasi-Newton updates to R involve $O(s^2)$ arithmetic operations, as do various other updates that are made to R as the active set changes.

In many cases s remains "moderate" (say less than 100) and the linear algebra involved in using and updating R is not excessive. For QP, this may be true if only a few of the variables are involved in the Hessian, or if the linear terms in the objective dominate the quadratic part. However, if the number of active constraints is substantially less than the number of variables, i.e., if s is large, the work involved in maintaining R is correspondingly substantial. In such cases we can expect alternative methods (such as a range-space approach or the barrier algorithm) to be superior.

8.10 The Test Environment

Both BARQP and MINOS are implemented in Fortran 77 and were run on one processor of an Alliant FX/8 with the UNIX operating system. The test runs were made as background jobs when the machine was lightly loaded. Although the Alliant has virtual memory, sufficient real memory was available to prevent significant paging activity. The recorded CPU times should be accurate to within about 1 per cent. Times are measured in CPU seconds.

For consistency we have scaled the constraint matrix A using the same scaling routine as in MINOS. It may seem that all of the KKT system should be scaled. However, such a scaling would have to be applied at every iteration (since $H + \mu D$ changes). Notice that scaling A does imply a symmetric scaling of H.

The problems were solved without any preprocessing of the data other than scaling. Most results reported in the literature for barrier methods are on problems that have been preprocessed, a procedure that apart from reducing the size of the problem can significantly improve the condition of the constraint matrix. Although preprocessing may be a sensible first step in practice, our purpose has been to try and test the limits of our method.

8.11 Quadratic Test Problems

A large collection of sparse linear programs resides in the *lp/data* chapter of *netlib* [Gay85]. The QP test problems were obtained from linear programs in this collection

	Rows	Columns	Nonzeros
SHARE2B	97	79	730
SHARE1B	118	225	1182
SCFXM1	331	457	2612
E226	224	282	2767
SCAGR25	472	500	2029
SHELL	537	1775	4900
SCTAP1	301	480	2052
SCSD1	78	760	3148
SCSD6	148	1350	5666

Table 8.1: LP test problems from *netlib*

by introducing a sparse matrix H into the objective. Initially we planned to set $H = \rho I$ for some fixed positive scalar ρ, but the choice of ρ was soon found to be problematical because of the widely varying scale of the vector c. For any given ρ, the quadratic objective

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^T x + \tfrac{1}{2}\rho x^T x$$

is sometimes essentially the same as $c^T x$, while in other cases it is dominated by the term $\rho x^T x$.

One solution would have been to set ρ to be a constant multiple of $\|c\|$. For simplicity we chose to ignore c and to set $\rho = 1$, thus obtaining the "minimum-length" objective

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad x^T x.$$

This is one of the simplest possible quadratic objective functions, but it provides a well-posed problem that often has physical meaning. Recall that the objective for any convex QP may be converted to this form (with some help of some additional constraints and variables), though only some of the variables then appear in the transformed objective.

Table 8.1 gives details of the constraint matrix A for a representative set of the QP test problems that were solved. Table 8.2 summarizes the performance of MINOS and BARQP on these problems. The iteration counts for both methods are included for interest, but of course the work per iteration differs substantially. Only the execution times should be compared.

The column headed s shows the "degrees of freedom" in the solution obtained

QP Problems		MINOS		BARQP	
Name	s	Itns	Time	Itns	Time
SHARE2B	6	116	3.4	31	23.3
SHARE1B	7	228	8.9	43	32.9
SCFXM1	34	509	35.2	37	144.9
E226	72	832	53.0	41	182.1
SCAGR25	95	610	62.6	30	49.3
SHELL	188	762	119.5	37	72.9
SCTAP1	189	821	89.0	34	39.2
SCSD1	303	955	207.6	25	77.8
SCSD6	528	1707	1930.0	32	94.5

Table 8.2: Comparison of MINOS and BARQP on QP test problems

by MINOS, as measured by the final number of superbasic variables. This is the dimension of the subspace in which MINOS applies a quasi-Newton method for unconstrained minimization, using a dense triangular matrix of dimension s.

It can be seen that as s grows, the efficiency of MINOS degrades substantially. In contrast, the efficiency of BARQP appears to depend only on the dimensions of A. We can conclude that barrier algorithms typified by BARQP are likely to be more efficient than conventional active-set reduced-gradient algorithms (typified by MINOS) on the class of QP problems having many degrees of freedom.

Regarding linear programs, several examples in the collection were chosen to provide realistic LP test problems. The implementation is capable of processing the larger *netlib* examples, but the execution time would be correspondingly greater. The iteration counts for BARQP are comparable to those obtained by primal barrier algorithms elsewhere (e.g. [GMS*86,Mar89]).

In most of the LP problems tried in our experimentation, the computation times for BARQP were higher than for the simplex implementation in MINOS. The explanation for this behavior lies in our use of the KKT system to obtain search directions, and the use of MA27 to solve each system. We anticipate the BARQP times might be reduced significantly by the use of MA47. For LP problems, H_B is diagonal and virtually all existing implementations reduce the KKT system to the Schur-complement form $AD^2A^T\Delta\pi = d$, where $D^2 = H_B^{-1}$ is diagonal. In many practical cases, AD^2A^T and its Cholesky factors are very sparse and can be computed efficiently with state-of-the-art software. Our use of the full KKT system rather than AD^2A^T can be expected

to provide greater reliability.

8.12 Sparsity in H

The sparsity of H affects the two algorithms in different ways. In MINOS the only requirement is to evaluate Hx and x^THx at each trial point. If H were in fact *dense*, the work would increase with problem size as a function of n^2, whereas in BARQP it would be of order n^3.

Thus without experimentation we can say that a large dense H may be more efficiently handled by MINOS, unless as before it leads to a large number of degrees of freedom. In general, increased sparsity in H helps both methods.

8.13 Conclusions

The purpose of the experiments has not been to conclude that BARQP in its present state is a highly efficient algorithm. Our goals have been more modest. Firstly, it is comforting that the number of iterations taken on the QP problems is similar to those taken on the LP problems. If a development of BARQP is to be competitive with active-set methods for QP then this is an essential requirement. Secondly, the CPU times for BARQP are in the same ballpark as those of the active-set method MINOS. Undoubtedly a special-purpose QP active-set method may perform rather better than MINOS, especially on a problem with many degrees of freedom. It is also highly likely that the efficiency of BARQP can be improved dramatically. A comparison of such codes must await both the development of BARQP and the provision of a large-scale active-set QP code. While such codes do exist the current approaches are not based on the symmetric factorization of the KKT matrix, which is likely to be the most efficient approach. An interesting characteristic of the barrier approach for QP is that the most effective methods for both the barrier algorithm and the active-set approach use the same matrix factorization. This should greatly facilitate numerical comparison.

At the moment the only approach advocated for the indefinite case is based on a primal algorithm. In the LP case it is known that primal-dual methods are the most effective approach. As described in Chapter 6, such an approach is possible in the case of convex QP; also see [CLMS90]. We could attempt such an approach in the indefinite case if we could be assured that at every iteration the KKT matrix has the required inertia. Failing that we would need to develop methods for computing appropriate directions of negative curvature.

At the commencement of this work it was uncertain that the barrier approach to LP could be extended to more general problems. It was known that the complexity results did extend to the convex QP case, but this was of little comfort. The concern

was whether the *numerical* performance of such algorithms was inherently flawed, since it was known that the barrier subproblems are in general ill-conditioned. (LP was the exception to this rule.) We can now assert that provided the barrier transformation is applied to a suitably formulated problem, there is no *a priori* reason for assuming that the performance of barrier algorithms on general problems is any different from that for the LP case. We anticipate that both barrier algorithms and active-set methods will play a significant role in solving large-scale nonlinear optimization problems.

Bibliography

[Aas71] J. O. Aasen. On the reduction of a symmetric matrix to tridiagonal form. *BIT*, 11:233–242, 1971.

[Arm66] L. Armijo. Minimization of functions having lipschitz-continuous first partial derivatives. *Pacific J. Math.*, 16:1–3, 1966.

[Bea55] E. M. L. Beale. On minimizing a convex function subject to linear inequalities. *J. Royal. Stat. Soc.*, 17B:173–184, 1955.

[Bea59] E. M. L. Beale. On quadratic programming. *Naval Research Logistics Quarterly*, 6:227–243, 1959.

[Bes84] M. J. Best. Equivalence of some quadratic programming algorithms. *Mathematical Programming*, 30:71–87, 1984.

[Big75] M. C. Biggs. Constrained minimization using recursive quadratic programming: some alternative subproblem formulations. In L. C. W. Dixon and G. P. Szegö, editors, *Towards Global Optimization*, pages 341–349, North-Holland, Amsterdam, 1975.

[BK77] J. R. Bunch and L. C. Kaufman. Some stable methods for calculating inertia and solving symmetric linear equations. *Mathematics of Computation*, 31:163–179, 1977.

[BK80] J. R. Bunch and L. C. Kaufman. A computational method for the indefinite quadratic programming problem. *Linear Algebra and its Applications*, 34():341–370, 1980.

[BP71] J. R. Bunch and B. N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM J. Numerical Analysis*, 8:639–655, 1971.

[Car59] C. W. Carroll. *An Operations Research Approach to the Economic Optimization of a Kraft Pulping Process.* PhD thesis, Institute of Paper Chemistry, Appleton, Wisconsin, 1959.

[Car61] C. W. Carroll. The created response surface technique for optimizing nonlinear restrained systems. *Operations Research*, 9:169–184, 1961.

[CD68] R. W. Cottle and G. B. Dantzig. Complementary pivot theory of mathematical programming. In G. B. Dantzig and A. F. Veinott, Jr. editors, *Mathematics of the Decision Sciences, Part I*, pages 115–136. American Mathematical Society, Providence, RI, 1968.

[CD79] R. W. Cottle and A. Djang. Algorithmic equivalence in quadratic programming, I: a least-distance programming problem. *Journal of Optimization Theory and Applications*, 28:275–301, 1979.

[CGLN84] E. Chu, A. George, J. Liu, and E. Ng. *SPARSPAK: Waterloo Sparse Matrix Package, User's Guide for SPARSPAK-A.* Technical Report CS-84-36, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1984.

[CLMS90] T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D. F. Shanno. *A primal-dual interior-point method for convex separable nonlinear programs.* Technical Report SOR 90-2, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, 1990.

[Con80] L. B. Contesse. Une caractérisation compléte des minima locaux en programation quadratique. *Numerische Mathematik*, 34:315–332, 1980.

[Cot68] R. W. Cottle. The principal pivoting method of quadratic programming. In G. B. Dantzig and A. F. Veinott, Jr, editors, *Mathematics of the Decision Sciences, Part I*, pages 144–162, American Mathematical Society, Providence, RI, 1968.

[Cot89] R. W. Cottle. *The principal pivoting method revisited.* Technical Report SOL 89-3, Department of Operations Research, Stanford University, Stanford, CA, 1989.

[Dan61] G. B. Dantzig. *Quadratic programming: A variant of the Wolfe-Markowitz algorithms.* Technical Report 2, Operations Research Center, University of California, Berkeley, CA, 1961.

[Dan63] G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, 1963.

[Den57] J. B. Dennis. *A dual problem for a class of quadratic programming problems.* Technical Report 1, M.I.T. Res. Note, Cambridge, MA, Nov. 1957.

[Den59] J. B. Dennis. *Mathematical Programming and Electrical Networks.* Technology Press of the Massachusetts Institute of Technology and John Wiley & Sons, Cambridge, MA, 1959.

[DES82] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM Journal on Numerical Analysis*, 19:400–408, 1982.

[DGR*89] I. S. Duff, N. I. M. Gould, J. K. Reid, J. A. Scott, and K. Turner. *The factorization of sparse symmetric indefinite matrices.* Technical Report CCS 236, Computer Science and Systems Division, AERE Harwell, Oxford, England, 1989.

[Dja79] A. Djang. *Algor. ..ic Equivalence in Quadratic Programming.* PhD thesis, Department of Op. ..tions Research, Stanford University, CA, 1979.

[DR82] I. S. Duff and J. K. Reid. *MA27: A set of Fortran subroutines for solving sparse symmetric sets of linear equations.* Technical Report AERE R-10533, Computer Science and Systems Division, AERE Harwell, Oxfordshire, England, 1982.

[DR83] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, 9:302–325, 1983.

[DS83] J. E. Dennis, Jr and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization.* Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[Fat78] Y. Fathi. *Computational complexitiy of LCP's associated with positive definite symmetric matrices.* Technical Report 78-2, Department of Industrial Engineering, University of Michigan, 1978.

[FF77] R. Fletcher and T. L. Freeman. A modified Newton method for minimization. *J. Optimization Theory and Applications*, 23:357–372, 1977.

[FGM89a] A. L. Forsgren, P. E. Gill, and W. Murray. *A modified Newton Method for Unconstrained Minimization.* Technical Report SOL 89-12, Department of Operations Research, Stanford University, CA, 1989.

[FGM89b] A. L. Forsgren, P. E. Gill, and W. Murray. *On the identification of local minimizers in inertia-controlling methods for quadratic programming.* Technical Report SOL 89-11, Department of Operations Research, Stanford University, CA, 1989. (To appear in *SIAM J. on Matrix Analysis and Applications*).

[Fle71] R. Fletcher. A general quadratic programming algorithm. *J. Institute of Mathematics and its Applications*, 7:76–91, 1971.

[Fle87] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, New York and Toronto, 1987.

[FM68] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* Wiley, New York, 1968.

[FM90] A. L. Forsgren and W. Murray. *Newton methods for large-scale linear equality-constrained minimization.* Technical Report SOL 90-6, Department of Operations Research, Stanford University, CA, 1990.

[Fri55] K. R. Frisch. *The logarithmic potential method of convex programming.* Technical Report, Memorandum of May 13, University Institute of Economics, Oslo, Norway, 1955.

[Gay85] D. M. Gay. Electronic mail distribution of linear programming test problems. *Mathematical Programming Society COAL Newsletter*, 13:10–12, 1985.

[GGM*84] P. E. Gill, N. I. M. Gould, W. Murray, M. A. Saunders, and M. H. Wright. A weighted Gram-Schmidt method for convex quadratic programming. *Mathematical Programming*, 30:176–195, 1984.

[GGMS74] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders. Methods for modifying matrix factorizations. *Mathematics of Computation*, 28:505–535, 1974.

[GI83] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27:1–33, 1983.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: a guide to the theroy of NP-completeness.* W. H. Freeman, San Francisco, CA, 1979.

[GKT51] D. Gale, H. W. Kuhn, and A. W. Tucker. Linear programming and the theory of games. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 317–329, John Wiley, New York, 1951.

[GL81] J. A. George and J. W. H. Liu. *Computer solution of large sparse positive definite systems.* Prentice-Hall Inc., Englewood Cliffs, NJ, 1981.

[GL89] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, Maryland, 1989.

[GM74] P. E. Gill and W. Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Mathematical Programming*, 7:311–350, 1974.

[GM78] P. E. Gill and W. Murray. Numerically stable methods for quadratic programming. *Mathematical Programming*, 14:349–372, 1978.

[GMS*86] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Mathematical Programming*, 36:183–209, 1986.

[GMSW84a] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Software and its relationship to methods. In P. T. Boggs, R. H. Byrd, and B. Schnabel, editors, *Proceedings of the SIAM Conference on Numerical Optimization*, pages 139–159, SIAM, Philadelphia, 1984.

[GMSW84b] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Sparse matrix methods in optimization. *SIAM J. on Scientific and Statistical Computing*, 5:562–589, 1984.

[GMSW84c] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. *User's Guide for QPSOL: A Fortran Package for Quadratic Programming.* Technical Report SOL 84-6, Department of Operations Research, Stanford University, Stanford, CA, 1984.

[GMSW85] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Model building and practical aspects of nonlinear programming. In K. Schittkowski, editor, *Computational Mathematical Programming*, pages 209–247, Springer-Verlag, Berlin, 1985.

[GMSW87] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. *A Schur-complement method for sparse quadratic programming.* Technical Report SOL 87-12, Department of Operations Research, Stanford University, Stanford, CA 1987. To appear in M. G. Cox and S. Hammarling (eds.), *Reliable Numerical Computation*, Oxford University Press.

[GMSW88] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. *Inertia-controlling methods for quadratic programming.* Technical Report SOL 88-3, Department of Operations Research, Stanford University, Stanford, CA, 1988. (To appear in SIAM Review).

[GMW81] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, London and New York, 1981.

[Gol67] A. A. Goldstein. *Constructive Real Analysis.* Harper & Row, New York, 1967.

[Gol80] D. Goldfarb. Curvilinear path steplength algorithms for minimization which use directions of negative curvature. *Mathematical Programming*, 18:31–40, 1980.

[Gon87] C. C. Gonzaga. *Search directions for interior linear programming methods.* Technical Report UCB/ERL M87/44, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA, 1987.

[Gou85] N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions of the general equality quadratic programming problem. *Mathematical Programming*, 32:90–99, 1985.

[Gou86a] N. I. M. Gould. *An algorithm for large-scale quadratic programming.* Technical Report CSS 219, Computer Science and Systems Division, AERE Harwell, Oxfordshire, OX11 1RA, England, 1986.

[Gou86b] N. I. M. Gould. On the accurate determination of search directions for simple differentiable penalty functions. *IMA Journal on Numerical Analysis*, 6:357–372, 1986.

[HS52] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.

[IBM76] IBM World Trade Corporation. *IBM Mathematical Programming System Extended/370 (MPSX/370) Program Reference Manual. 2nd ed. No. SH19-1095-1*, New York and Paris, 1976.

[Kar74] R. M. Karp, editor. *Complexity of Computation.* Volume VII of *SIAM-AMS Proceedings*, American Mathematical Society, Providence, Rhode Island, 1974. Held in NY, April 1973.

[Kar84] N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[KD79] S. Kaniel and A. Dax. A modified Newton's method for unconstrained minimization. *SIAM J. on Numerical Analysis*, 16:324–331, 1979.

[KM72] V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Proceedings of the Third Symposium on Inequalities*, pages 159–175, Academic Press, New York, 1972.

[KMNY90] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A unified approach to interior-point algorithms for linear complementarity problems.* Technical Report RJ 7493 (70008), Mathematics/Computer Science, IBM Research Division, 1990.

[Lem62] C. E. Lemke. A method of solution for quadratic programs. *Management Science*, 8:442–453, 1962.

[LMS89] I. J. Lustig, R. E. Marsten, and D. F. Shanno. *Computational experience with a primal-dual interior-point method for linear programming.* Technical Report SOR 89-17, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, 1989.

[LMS90] I. J. Lustig, R. E. Marsten, and D. F. Shanno. *On implementing Mehrotra's predictor-corrector interior-point method for linear programming.* Technical Report SOR 90-03, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, 1990.

[Lus89] I. J. Lustig. *An analysis of an available set of linear programming test problems.* Technical Report SOL 87-11, Department of Operations Research, Stanford University, CA, 1989.

[Maj71] A. Majthay. Optimality conditions for quadratic programming. *Mathematical Programming*, 1:359–365, 1971.

[Mar89] A. Marxen. *Primal Barrier Methods for Linear Programming.* PhD thesis, Department of Operations Research, Stanford University, CA, 1989. Report SOL 89-6.

[McC77] G. P. McCormick. A modification of Armijo's step-size rule for negative curvature. *Mathematical Programming*, 13:111–115, 1977.

[Meh89] S. Mehrotra. *On finding a vertex solution using interior-point methods.* Technical Report 89-22, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1989.

[Meh90] S. Mehrotra. *On the implementation of a (primal-dual) interior point method.* Technical Report 90-03, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1990.

[MK87] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear complementarity problems. *Mathematical Programming*, 39:117–129, 1987.

[MLB87] C. Moler, J. Little, and S. Bangert. *PRO-MATLAB User's Guide.* The MathWorks, Inc., Sherborn, MA, 1987.

[MP78] H. Mukai and E. Polak. A second-order method for unconstrained optimization. *J. Optimization Theory and Applications*, 26:501–513, 1978.

[MS78] B. A. Murtagh and M. A. Saunders. Large-scale linearly constrained optimization. *Mathematical Programming*, 14:41–72, 1978.

[MS79] J. J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Mathematical Programming*, 16:1–20, 1979.

[MS83] B. A. Murtagh and M. A. Saunders. *MINOS 5.0 User's Guide.* Technical Report SOL 83-20, Department of Operations Research, Stanford University, Stanford, CA, 1983.

[Mur69] W. Murray. *Indefinite quadratic programming.* Technical Report Ma 76, National Physical Laboratory, Teddington, Middlesex, England, 1969.

[Mur71a] W. Murray. *An algorithm for finding a local minimum of an indefinite quadratic program.* Technical Report NAC 1, National Physical Laboratory, Teddington, Middlesex, England, 1971.

[Mur71b] W. Murray. Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions. *J. Optimization Theory and Applications*, 7:189-196, 1971.

[Pow81] M. J. D. Powell. An upper-triangular matrix method for quadratic programming. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 4*, pages 1–24, Academic Press, London and New York, 1981.

[PS82] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice Hall, Englewood Cliffs, NJ, 1982.

[PS88] P. M. Pardalos and G. Schnitger. Checking local optimality in constrained quadratic programming is NP-hard. *Operations Research Letters*, 7:33–35, 1988.

[RV88] C. Roos and J.-Ph. Vial. *Long steps with the logarithmic penalty barrier function in linear programming.* Technical Report 89-44, Faculty of Mathematics and Informatics/Computer Science, Delft University of Technology, Delft, Holland, 1988.

[SAM80] B. Stott, O. Alsac, and J. L. Marinho. The optimal power flow problem. In A. M. Erisman, K. W. Neves, and M. H. Dwarakanath, editors, *Proceedings of SIAM Conference on Electric Power Problems: The Mathematical Challenge*, pages 327–351, SIAM, Philadelphia, 1980.

[SMD86] J. C. Stone, P. H. McAllister, and G. B. Dantzig. *Using the PILOT model to study the effects of technological change.* Technical Report SOL 86-16, Department of Operations Research, Stanford University, Stanford, CA, 1986.

[TMY90] M. Todd, S. Mizuno, and Y. Ye. Anticipated behavior of path-following algorithms for linear programming. Presented at the Second Asilomar Workshop on Progress in Mathematical Programming, February 1990.

[Tom76a] J. A. Tomlin. *Programmer's Guide to LCPL: a program for solving linear complementarity problems by Lemke's method.* Technical Report SOL 76-24, Department of Operations Research, Stanford University, Stanford, CA, 1976.

[Tom76b] J. A. Tomlin. *Robust implementation of Lemke's method for the linear complementarity problem.* Technical Report SOL 76-24, Department of Operations Research, Stanford University, Stanford, CA, 1976.

[Tom76c] J. A. Tomlin. *User's Guide to LCPL: a program for solving linear complementarity problems by Lemke's method.* Technical Report SOL 76-16, Department of Operations Research, Stanford University, Stanford, CA, 1976.

[VL90] R. J. Vanderbei and J. C. Lagarias. I. I. Dikin's convergence result for the affine-scaling algorithm. *Contemporary Math*, 1990. To appear.

[Wol59] P. Wolfe. The simplex method for quadratic programming. *Econometrica*, 27:382–398, 1959.

SOL 91-2: Barrier Methods for Large-Scale Quadratic Programming, Dulce B. Ponceleón (June 1991, 117 pp.).

We present several new algorithms for solving the general large-scale quadratic programming (QP) problem.

A feature of QP problems is the presence of linear inequality constraints, which introduce a combinatorial aspect to the problem. Currently the most common approach to solving QP problems is to apply active-set methods, in which only some of the inequalities are used to produce a search direction at each stage The combinatorial element is therefore inherent. As problems become larger, there is a potential for an excessive number of iterations and consequent inefficiency.

In contrast, we use the now familiar barrier-function approach, which circumvents the combinatorial aspect by introducing a barrier transformation involving all of the inequalities. The barrier term enforces satisfaction of the inequalities by modifying the objective function. The transformed problem is solved by a modified Newton Method applied to the nonlinear equations defining feasibility and optimality.

The main computation at each iteration of the new algorithms is the solution of an indefinite system of linear equations. Barrier methods are known to lead to ill-conditioned systems. However, we show by a special sensitivity analysis that the particular manner in which we have formualted the barrier transformation leads to ill-conditioning that is benign.

We address the many details that need to be resolved in order to define an efficient algorithm for solving large-scale QP problems. A specific barrier algorithm has been implemented, with linear programming (LP) included as a special case. Numerical results are presented for a set of sparse QP test problems. A feature of the implementation is that its efficiency does not depend on whether the solutuion is near or far from a vertex of the feasible region.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1991	Technical Report

4. TITLE AND SUBTITLE
Barrier Methods for Large-Scale Quadratic Programming

5. FUNDING NUMBERS
N00014-90-J-1242

6. AUTHOR(S)
Dulce B. Ponceleón

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Operations Research - SOL
Stanford University
Stanford, CA 94305-4022

8. PERFORMING ORGANIZATION REPORT NUMBER
1111MA

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research - Department of the Navy
800 W. Quincy Street
Arlington, VA 22217

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
SOL 91-2

12a DISTRIBUTION AVAILABILITY STATEMENT
UNLIMITED

12b. DISTRIBUTION CODE
UL

13. ABSTRACT (Maximum 200 words)

(see reverse side)

14. SUBJECT TERMS
linear programming; quadratic programming; indefinite systems; KKT systems; barrier methods; interior-point methods.

15. NUMBER OF PAGES
117 pp.

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED			UL